advice. In fact, consumers typically configure their advice reader to subscribe mainly to advice from large concerns which manufacture goods and services of interest to the consumer such as, for example, a computer manufacturer, a software publisher, or the provider of Internet service. Subscription to substantial organizations of this type is a reasonably secure practice. Such organizations have an interest in providing trustworthy advice so that they maintain rapport with their consumers. It is anticipated that very few risks are posed to advice consumers who subscribe to advice authored by such concerns.

- Better Advice Bureau. The Better Advice Bureau.org, which is described above, is a fundamental tool for ensuring the security of invention users. All invention users subscribe to this site. This site compiles counter advice, informing users about dangerous sites and about bad advice which is circulating. The Better Advice Bureau functions in some respects as an immune system for the invention, allowing the correction of dangerous situations. UrgentAdviceNet is another site to which all users subscribe. It provides a special mechanism for delivering very urgent counter advice to the consumer population.

Absence of High Profile Risk

The following discussion of security considers some of the more well known risks of Internet interaction and then explains why these well known risks actually do not arise under the invention when used in a typical implementation.

Inventory of High Profile Risks

Internet operations have in the past suffered a number of active threats that can be symbolized by three figures who have captured the popular imagination:

- Break-ins: Kevin Mitnick. Over a period of years Mitnick used the Internet systematically to break into computers worldwide, and he managed deliberately to cause some to crash or to lose data permanently. While it is supposed that Mitnick was some sort of evil genius the truth is that sites on the Internet give instructions on how to break into Pentagon computers. A Pentagon led experiment in 1997 showed that using publicly available information one could, in fact, access classified DOD computers and cause permanent damage to files.

- Attacks. The Internet currently makes software tools available for free which allow their users to attack other peoples computers over the Internet, causing those computers to crash. The basic strategy is to connect to various TCP/IP port servers on the intended victim computer and flood it with requests for service. (Anonymous, Maximum Security, Sams.Net 1997)

- Worms: Robert Morris, Jr. In a well-known 1988 episode, Morris released a worm which spread rapidly across the Internet, installing itself in many machines, and while in execution on those machines, spread itself to other machines. In fact, Morris was attempting no more than a prank. The rapid and pervasive spread of the worm surprised him, as did the enormous amount of time required to eradicate the worm and regain full capabilities of the affected computers. The powerfully disruptive nature of the worm was caused by its ability to spread automatically, and run automatically on whatever machine it reached. This case dramatizes the risks that can arise through the automatic spreading of executable code across the Internet. (Pfleeger, Security in computing, Prentice Hall 1996)

Absence of Consumer Exposure to High-Profile Risk

The advice reader does not expose the consumer to additional risk from these high profile sources beyond the baseline risk he suffers now.

The advice reader is not vulnerable to break-in because it does not offer any kind of interactive shell offering log-in access, as the term break-in requires.

The advice reader does not expose the consumer computer to any extra risk of attack beyond the risk the consumer already faces due to Internet connectivity.

The advice reader adds no risk because it does not make available any perpetually open TCP/IP port which can be flooded with requests. There is nothing the outside world can do to try to talk to or initiate an interaction with the advice reader.

The advice reader does not expose the network to any risks of worms. In a typical configuration, the system does not offer any mechanism by which anything can spread from advice reader to advice reader.

Server Exposure

Consider the vulnerability of the invention server to active threats. A server using the invention, as with any Internet-based server, exists for the purpose of offering services to the outside world. It is visible on the Internet and open for business, typically around the clock.

There is no risk of break-in, because there is no interactive shell offering log-in access, as the term break-in implies. However, the server can be flooded with requests as with any Internet server. There are well known techniques to combat such request floods, and professional Web site operators know about them. The server side users of the invention are professionals who are well equipped to evaluate and react to this type of standard threat.

The invention's server does not expose the server to any risks of worms. In a typical configuration, the system does not offer any mechanism by which anything can spread from advice reader to advice server, or by which anything other than an extremely narrow range of functions can be performed by the server.

Protective Influence

There is a certain sense in which the invention actually can help protect against worms, break-ins, and attacks. The advice delivery mechanism allows network security personnel to create advisories warning the consumer when the consumer is behaving in a way that leaves the door open to criminal disruption. The advice delivery mechanism also allows network security personnel to author advisories which diagnose whether a user is currently being attacked, or has been recently attacked. In this way, the invention functions as an immune system, allowing the rapid spread of corrective advice.

Spoofing Risks

In effect, the invention interaction is never completely unsupervised. The advice reader only interacts with advice sites that have been subscribed to by the user. The user is therefore, in his choice of subscriptions, exerting a kind of permanent high level supervision. If the user subscribes only to sites offered by organizations with a strong incentive to provide trustworthy advice, he is protected. An individual making harmful advice does not legally have a way to force the introduction of that advice into any given advice reader.

There is a very important category of active threat which is not widely known, *i.e.* attack by spoofing. In this category falls spoofing of Internet locations, *i.e.* the user thinks he is communicating with a certain trusted site, but actually is communicating with an impostor site. Another kind of spoofing is the use of mole programs which appear to be standard applications but which actually are not, and can violate privacy and security in other ways. (Anonymous, Maximum Security, Sams.Net 1997)

DNS Spoofing

In this scenario, an impostor creates a near clone of a popular and trusted site, such as the site of MicroComp. However, the impostor site also contains harmful advice.

DNS spoofing provides a way for the impostor site to appear to certain users on the network as if it were actually the popular and trusted site of MicroComp. The only way this could happen under current network protocols is for the impostor to interfere with the DNS lookup process of certain consumers, and misdirect certain consumer advice requests aimed for MicroComp.

DNS spoofing operates as follows: The impostor must have system level access to a machine on the Internet which is physically located in a position to intercept some of the domain name resolution requests intended for a certain Domain Name Server (DNS). The impostor programs the IP routing logic to inspect the intercepted requests looking for those which refer to MicroComp and, when such a request is found, to return an incorrect TCP/IP address, the returned address referring to his fake advice site. All advice readers situated downstream from the impostor are in this way misdirected to the fake advice site whenever they try to go to the MicroComp advice site. The fake site appears just like a real site, but distributes harmful advice under the pretense of being a trusted provider. In short, by perpetrating DNS fraud, there is a way for an attacker to introduce damaging advice directly into one or many computers.

This sort of activity constitutes criminal fraud under current federal regulations. This type of fraud is reportedly rare (see Anonymous (1997) Maximum Security, Sams.net Publishing, Indianapolis. In addition, a perpetrator able to carry off this type of fraud might find systems using the invention to be less attractive than other targets. For example, DNS spoofing of large electronic commerce sites such as bookstores and computer software warehouses is more attractive to the perpetrator, in the sense of offering a more rewarding payoff if the spoof is successful. Indeed, the perpetrator could offer a Web site pretending to be the Web site of a certain merchant, offering up Web pages with the same general visual appearance as Web pages from the correct site. The fake Web site contains forms which the user fills out to execute the transaction. In reality, those forms are used to capture information about credit card numbers or other sensitive financial data. This seems a more direct way for a perpetrator to benefit from a DNS spoofing scheme.

This sort of activity affects only a subset of the users of a large public network such as the Internet. For example, assuming that an individual consumer enjoys a secure connection to a DNS server, and assuming also that the information on the DNS is maintained securely, DNS spoofing is not a material threat for that particular consumer. In most moderately large corporate environments, DNS services are provided within the corporate intranet. Assuming that the impostor is outside the corporation, then for advice consumers within the corporation, this spoofing threat is stymied by the standard security devices for intranets, *i.e.* firewalls. Certain noncorporate advice consumers enjoy Internet access through Internet service providers offering DNS servers located on the Internet in close physical proximity to their modem banks. Assuming that the impostor is not inside the physical domain of the Internet service provider's offices, consumers who use such DNS services may also be secure against DNS spoofing.

In effect, spoofing is only a threat for advice readers relying on insecure connections to their DNS. In future network protocols, DNS connections may be digitally authenticated, and the spoofing threat is stymied in such settings as well. Until that time, the invention has a way to stymie this threat under the current regime using digital authentication of advice itself. Digital authentication of advice is also of interest to those consumers with secure DNS connections because advice may be distributed, in some implementations, by insecure means such as e-mail or sneakernet. It gives the user additional confidence in the advice he is receiving.

In a typical implementation of invention, the term digital authentication refers to the use of existing digital signature mechanisms based on so called public-key/private-key pairs (see PGP 4.0 Users Manual, PGP Pretty Good Privacy, Inc. (1997)). This mechanism is developing into a well understood, mature, and reliable standard. Other forms of digital authentication can be used with equal validity.

The following describes how the public-private key pair mechanism is used to authenticate advice. The advice provider, *e.g.* MicroComp, acquires a public-key/private-key pair, of which the private key is a secret known only to the provider. The provider takes steps, described below, to publicize the correct public key. The provider, knowing both keys of the pair, attaches to each advisory a signature block which is successfully interpreted by an advice reader which knows the correct public key. The ability to interpret the block is considered by the advice reader proof that the author knew both keys, which is considered proof that the author is in fact MicroComp. In a typical implementation, a user interface component informs the user that a given piece of advice is signed by MicroComp. The precise meaning of this is that the signature block is successfully interpreted by using the known public key.

The invention's mechanism for protection from the DNS spoofing threat involves actions by both the consumer and the provider. The provider authors a site description file, containing a listing of the information related to the subscription, including the site's location and the site's digital signature public key. The provider publishes the site description file, for example in physical media such as a disk or CD-ROM, as part of the distribution of a software product offered by MicroComp. In this way, many consumers obtain copies of the site description file by secure means. A consumer initializing a subscription to MicroComp presents to the advice reader's subscription manager the site description file for MicroComp. The provider, whenever authoring an advisory, attaches a digital signature block. The advice reader, whenever obtaining a piece of advice, checks that the digital signature is successfully interpreted using the public key known to the reader to correspond to MicroComp. Unless the advisory passes this test, the advice reader refuses to evaluate the advice for relevance. The reader may also notify the user that there is unsigned advice coming from a site whose site description file claims that the site provides only signed advice. The reader also offers to inform Better Advice Bureau of this fact.

To see why this approach protects against DNS spoofing, it is important to understand a basic feature of the public-key/private-key system. It is commonly accepted that an impostor faces a very difficult time trying to fake the digital signature of MicroComp.Com. This conclusion rests on the assumption that the impostor must make a successful fake signature using only the publicly available information associated with the encryption scheme; *i.e.* that the impostor does not have access directly to MicroComp.Com's private key. It is computationally an extremely difficult task for an impostor to fake a digital signature correctly from publicly available data (see C. Pfleeger, Security in Computing, Second Edition, Prentice-Hall(1996); and PGP 4.0 Users Manual, PGP Pretty Good Privacy, Inc. (1997)). It is an equivalent computational task to the task of factoring an integer with hundreds or thousands of digits into its prime factors. Using networks of many thousands of computer workstations over periods of many months, it has been possible to factor individual numbers with about 150-200 digits. However, this has been achieved only by a kind of vast scientific collaborative enterprise. It is unlikely that an impostor has access to the required resources for mounting an effort that would succeed on integers of the lengths commonly used in signature algorithms. Moreover, there is an easy remedy, *i.e.* double the number of digits of the keys, putting the factorization task beyond reach of any currently conceivable collaborative effort based on currently conceivable computational resources.

In short, an impostor is highly unlikely to be able to author advice with a digital signature which is intelligible using the correct MicroComp public key. Unless the impostor can do this, the advice reader refuses to evaluate the advice for relevance, and so the impostors advice poses no substantial threat.

Key Spoofing

An apparent hole in the digital authentication system is the possibility of key spoofing. In this scenario, the consumer's advice reader has somehow accepted an incorrect public key for MicroComp, *i.e.* a key which is not the correct key for MicroComp, but is instead the public key of a public-key/private-key pair owned by the impostor. If this happens, then the advice reader can be deceived because it recognizes the impostor's advice as valid. However, the invention is designed to prevent this scenario from occurring.

For key spoofing to occur, the consumer's subscription must be initiated using a site description file that is not obtained through secure channels, such as the original software installation from physical media. The impostor must author fake site description files and distribute these on the Internet.

A typical implementation of invention cannot be fooled by key spoofing. There are three mechanisms for this, any combination of which may be effective:

- Certification of site description files. In one implementation, site description files may include a digital signature by a central authority, the Better Advice Bureau, testifying that the site description file purporting to be authored by MicroComp is, in fact, so authored. The digital signature of Better Advice Bureau is hard wired into the advice reader, thereby avoiding the possibility of spoofing the Better Advice Bureau certification.

- Spoof-Proof Key Verification. A typical implementation of the subscription manager performs key verification prior to recording a subscription. It contains hard wired information enabling it to make a direct TCP/IP connection to a hard wired IP address of a key authentication server. Such a server verifies that a given organizations public key is as it is said to be. Because the contact address of the server is hard wired into the program, access to the key server cannot be DNS spoofed.

- Counter-advice. If a certain site is successfully spoofed, it may submit to Better Advice Bureau.org an advisory which goes out to all advice readers because Better Advice Bureau.org is a built-in subscription. The advisory asserts the value of the correct public key associated with the site. Those users with incorrect public keys are notified with the relevant advisory, which explains the risks involved. If the issue is particularly urgent, the site UrgentAdviceNet is employed.

In summary, if the advice reader and its subscriptions are appropriately configured, the advice consumer is protected from spoofing when the advice provider digitally signs his advisories.

Reduction of Spoofing Threats

DNS Spoofing, while a significant threat to Internet security, is not more of a threat to the invention than to other components of the Internet, especially e-commerce. The Better Advice Bureau.org and UrgentAdviceNet are important devices to help suppress spoofing of advice.

Better Advice Bureau.org and UrgentAdviceNet are important devices to help suppress spoofing of all Internet activities. By the use of this combination, the Internets susceptibility to spoofing may be reduced, and the attractiveness of spoofing in other settings, outside of invention are reduced.

<u>Advice Reader Moles</u>

Another potential hole in the invention's security system is the possibility that a copy of the executable binary of a legitimate advice reader is acquired by an attacker, and then is systematically altered to introduce various new behaviors. The resultant illegitimate reader is then redistributed on the Internet, where it masquerades as a legitimate copy of the advice reader, and is downloaded and used by unsuspecting consumers. Nothing can stop the creation of such illegitimate readers. Nothing can stop illegitimate versions of a software tool from displaying very damaging behavior. This is well understood by the community of Internet users worldwide. Anyone who downloads software over the Internet from sites which are not authentic providers of trusted software exposes himself to the same risk, whether the software is a word processor, a spreadsheet, a Web browser, or the advice reader.

However, of concern is the possibility of illegitimate mole readers whose goal is not to cause damage but to compromise the security and privacy of the user. Such mole readers contain subtle features escaping detection by casual observation but allowing for subtle effects on the user's environment or for the gathering and forwarding of important information about the user. Again, the invention is no more vulnerable to this kind of modification than any other piece of software. However, the typical implementation of the invention contains two mechanisms which can identify the existence of mole software and help correct the situation.

- Server-Challenge. This is implemented as part of the invention server-reader interaction protocol. A typical implementation of the server begins its transaction with an advice reader through a handshaking session, in which the server challenges the reader to prove that it is a valid version of an advice reader. In a typical implementation, the advice reader is written to create certain data blocks with known properties dynamically in memory at known location offsets from the beginning of the program. The method by which the data was created and the purpose of the creation are guarded secrets. The server selects random blocks of this data and asks the reader for the correct digital digest associated with such a block. If the program is altered, it is difficult for the executable code to answer the challenge correctly. If the server receives an unsatisfactory answer, the server then transmits advice to the reader which is automatically relevant, stating that the user's advice reader appears illegitimate. The advice reader may also refuse to interact with servers that do not pass a digital authentication test.

- Advice-Challenge. The invention, at Better Advice Bureau.org, offers advice whose intent is to verify that a valid configuration of the invention is installed. The advice, which may change daily, asserts that certain blocks of the data in the CPU memory while the advice reader is running have certain digital digests. The blocks are chosen randomly by the Better Advice Bureau.org authority, or according to design, when a certain well known mole is to be diagnosed from a specific motif in the binary data at a specific location.

In summary, invention diagnoses moles and notifies users about them.

Reduction of Mole Threats

Moles, while a potential threat to Internet security and privacy, are not more of a threat to the invention than to other components of the Internet, especially e-commerce. Better Advice Bureau.org and UrgentAdviceNet are important devices to help suppress spoofing. The same remark applies to moles. Better Advice Bureau.org and UrgentAdviceNet are important devices to help suppress mole applications uniformly. By the use of these devices, the Internet's susceptibility to mole activities may be reduced, and the attractiveness of mole activities in other settings, outside of invention is reduced.

Irreducible Core Risks

A threat is caused by defective advice offered in good faith by usually trustworthy authors. Advice authors have reputation incentives which tend to make them want to provide good advice. Advice providers in one core application, *e.g.* technical support, are part of sophisticated organizations which have the ability to do things in a disciplined way. They understand that advice should be tested for safety and effectiveness and be released in a deliberate, staged manner. Because of this, it is likely that very few pieces of advice in the technical support applications area are defective. Nevertheless, there are occasional problems with advice authored by typically trustworthy providers.

The risks posed by advice are of two kinds:

First, there are the risks posed by advice gathering and evaluation.

Second, there are risks posed by the solution process, *i.e.* by the users response to a relevant advisory which offers the user a solution to a problem. This second type of risk is by far the more serious one. When the user agrees to a solution, he is allowing powerful actions with potentially permanent consequences. The advice reader is not able to provide any kind of protection against the effects of applying flawed or malicious solutions. Instead, the burden of security must fall on the user, who should always limit subscriptions to well known, trusted sites, and should always carefully check the explanation and the authenticity of authorship before accepting a solution proposed by an advisory. In its typical configuration, invention does not automatically apply solution operators, precisely because of the need for user supervision.

As for the first kind of risk, that from gathering and evaluation, the invention is specially designed to limit risk.

It is true that the invention is typically used in a mode of automatic unattended operation. In this mode, advisories are gathered from external advice sites without user intervention and are automatically evaluated for relevance without user intervention. As mentioned earlier, the consensus of Internet experts is that automatic unattended operation over the Internet poses serious risks.

However, the invention does not download arbitrary resources, nor does it evaluate arbitrary executable code. Its design imposes constraints on what information can flow into the computer automatically, and on what effects automatic evaluation can have. These constraints are specifically imposed to avoid the known risks of unattended operation.

In its typical configuration, the invention does not automatically apply solution operators, even when performing automatic unattended operation. In that typical configuration, the effects of automatic unattended operation on the system are not direct effects, *i.e.* the advice reader does not enable modify access to a specific piece of the system environment. The effects are instead indirect, *i.e.* side effects of consuming too many resources during the downloading and evaluation of advice. The side effects to be concerned with are of three types:

(a) Advice gathering might monopolize all network bandwidth.

(b) Advice gathering might fill up the local storage device.

(c) Relevance evaluation might consume all CPU cycles.

Problems (a) and (b) are solved by resource rationing. The information that can flow into the computer consists of ASCII text files. By imposing resource quotas at download time, the system protects against the possibility that overly many network resources are used and protects against the possibility that overly big files are downloaded into the machine, exhausting the capacity of the processor or storage device. Problem (c) is also partly solved by resource rationing. By metering CPU usage and imposing resource quotas, the invention can address the problem.

Security Support in the Invention

The invention is designed to support security habits in several ways.

Language Structure

The relevance language is an example of mobile code. Such code is written by an author on one computer for interpretation on another computer. Recently, there has been considerable interest in the development of safe languages for mobile coding (see S. Oaks, Java Security, Oreilly(1998); and N. Borenstein, *Email with a mind of its own: The Safe-TCL Language for Enabled mail*, http://minsky.med.Virginia.edu:80/sdm7g/Projects/Python/safe-tcl/). Java and Safe-TCL are examples of so called safe languages, *i.e.* they are considered to provide a degree of safety that traditional languages such as C and C++ cannot offer.

The relevance language is a language for mobile coding. The language offers a level of security protection in excess of the current norm of the Internet business community. Relevance Language interpretation is inherently safer than safe languages for mobile code, such as Java and TCL. Java, TCL, and related languages are procedural languages. They contain control features such as loops, recursion, and branching statements which, if abused, can consume large fractions of system CPU resources. They offer authors storage allocation facilities which, if abused, can potentially consume large fractions of system memory resources. Remote unattended operation of code from these languages obtained over the Internet can in fact be dangerous, despite the labeling as safe. In fact, these mobile code languages are typically only used in attended operation. For example, mobile Java code is typically used in Web browsers, with a human watching the screen as the code runs. It is implicitly understood that the human is supervising the execution of the process.

The relevance language is a descriptive language rather than a procedural language. It describes a state of the computer and its environment. Relevance evaluation is a process of determining whether this state holds or not. This description of the state uses a language that does not exhibit traditional control structures, such as loops, nor does it have traditional storage allocation facilities.

In fact, the relevance language is so tightly constrained that it is not Turing-complete. It does not suffer from the famous Turing halting problem, which is a typical property of procedural languages. The Turing halting problem is to decide whether a given computer program ever halts or not. Most procedural languages are undecidable. They contain programs, perhaps even simple ones, for which it can never be known in advance whether the program must always halt. Java and TCL programs can be undecidable. In stark contrast, statements expressible in the relevance language are decidable, *i.e.* they halt.

This is an additional level of security that goes well beyond the security guarantees of mobile code languages, such as Java and TCL.

Human Intelligibility

An additional security feature of the invention is the human intelligibility of the relevance language. The relevance language has an appearance which is reminiscent of ordinary English. A consumer who reads English can form an approximate sense of what a given piece of advice is doing by inspecting the plain text of the advisory. In this way, consumers are brought into the process of understanding the advisories sent to them. While it is true that untrustworthy advice providers, by writing opaque relevance clauses, may still be able to disguise their intentions, the more important point is that trustworthy advice providers are able to make their intentions clear to consumers, and thereby gain and cultivate trust.

Disclosure and Labeling

The invention offers, in one implementation, a mechanism to encourage advice providers to label their advisories clearly for intended effects and thereby provide the public an accurate understanding of the risks associated with a given solution operators.

In this implementation, the Better Advice Bureau defines and maintains a list of special labels which indicate the effects of a certain solution operator, for example, the subsystems affected, the extent to which effects are reversible, and the availability of further documentation explaining the proposed change. The advice provider uses this labeling system to describe the effects of the advisories published by the provider. The advice reader uses this labeling mechanism as part of its user interface during the solution proposal process. When a consumer is contemplating applying a solution operator, part of the user interface indicates for the consumer the types of side effects which may result, according to the labeling which the provider has supplied.

Both consumers and providers, under the guidance of a central classification, come to have a common way to understand and discuss the potential effects of a system modification. The Better Advice Bureau issues counter advisories against advisories which inaccurately label the effects of their advisories. The advice reader uses distinctive visual identifiers to call attention to advice with extreme effects and to call attention to advice with no labeled effects. The consumer may refuse to approve proposed solution operators which are unlabelled, or to subscribe to sites which author unlabelled operators.

Security Summary

There are several illegal activities that threaten the security of the consumer. However, in every instance, the system has been designed with an effective means of defense. The invention does not expose the user to levels of risk in excess of those risks already experienced through the use of e-mail and Web browsing. In fact, the risks from invention are far lower than the risks of those standard activities.

There is also the possibility that otherwise trustworthy advice authors release damaging advice. The system is designed to contain and correct such situations. The extent of damage due to honest mistakes is contained because advice has access to only a limited complement of system resources, *e.g.* disk storage and CPU time, and the use of these resources is metered and rationed in a typical implementation. The structure of advice files and the associated relevance language is relatively transparent to consumers, which helps them play a role in fostering their own security. Finally, through the advisory process, through Better Advice Bureau and UrgentAdviceNet, the invention contains mechanisms to correct security problems automatically as they arise.

Privacy Issues

The advice reader accesses a great deal of information about the consumer's computer, about the contents of the files on the consumer's computer, and about the interactions of that computer with devices in the immediate environment. To the extent that the consumer stores information about his financial, personal, or medical affairs on the computer, typical implementations of the advice reader are able to access that information, for example bank balances and prescription drug information. To the extent that the consumer computer has access to network devices which form part of the consumer's home or work environment, the advice reader is able to access information about that environment, for example whether certain devices are present in the environment, whether they are operating, and what their conditions of operation are. Enabling the invention to access this information is beneficial to the consumer, allowing helpful advice to be written which can identify problematic situations and call them to the attention of the consumer.

Much of the information that invention has access to is potentially sensitive, and most consumers would not knowingly permit such data to be divulged to strangers. Any system which can access such sensitive information must also protect the information. As explained below, the advice reader acts to preserve the privacy of the consumer.

Existing Internet Privacy Standards

The invention is designed to protect user privacy, offering a level of protection far in excess of the current norm of the Internet business community.

Internet mediated activities, such as Web browsing and on-line commerce, can result in the disclosure to Web servers of information about the browsing consumer's identity, computer configuration, and also certain items about consumer shopping or browsing interests. There is no single accepted standard of privacy, and industry groups have formed for the purpose of gathering information about consumers from their Web interactions and sharing among themselves information about the consumers. Consumer oriented groups such as EPIC (Electronic Privacy information Center) have formed in response, and there are currently political battles over the consumer's right to electronic privacy.

The invention offers a method which meets or exceeds the level of information privacy desired by consumer groups, while providing the fine grained targeting of messages to recipients desired by industry groups.

The standard that the invention offers is understood by considering a classification of privacy respecting/threatening behaviors. The ethical standards of advice providers are classified into four categories, definitions of which are provided below.

(Ea) Completely Ethical

(Eb) Merely Ethical

(Ec) Merely Legal

(Ed) Criminal

Completely ethical behavior of an information provider is defined as full respect of consumer privacy and of the intended purpose of the invention communications protocol. A completely ethical provider would ...

- never seek to perform covert identification or surveillance of a consumer community. In particular, it would:

    make no efforts to infer from server activity the identity or attributes of any consumer,

    make no efforts to infer from network activity the attributes of any consumer, and

- make no efforts to use the Internet as a pure broadcast advertising medium, creating messages which make unsolicited contact with all or a very large number of consumers passively receiving messages.

- fully disclose to consumers the existence and purpose of data gathering efforts;

- make no efforts to use information so received in ways unrelated to the disclosed purpose of the information gathering effort;

- make no efforts to use information gathered from such a questionnaire to correlate with future server or network activity.

Completely ethical behavior is a standard much higher than that obeyed by many actors in the current Internet business community. The Internet business community at the moment contains a wide range of attitudes and behaviors towards consumer privacy. There are many instances of behavior that can be classified as merely ethical, or merely legal.

Merely ethical means that the behavior of inferring user identity or attributes from Internet activity, while providing some sort of notice that privacy compromises are taking place, respects the provider-consumer relationship by not using the information to initiate unwanted contacts with consumers and not sharing the information with other businesses. In effect, merely ethical behavior restricts the use of information gathering to internal research and planning purposes, in much the same way that ethical companies currently use information gathered from product registration cards.

Merely legal means that the behavior of inferring user identity or attributes from Internet activity, provides only minimal notice that some sort of privacy compromise is taking place, and then subsequently makes maximum exploitation of the gathered information under current laws, which includes systematically sharing the information with other businesses and initiating unwanted contacts with consumers. The standard of many Internet based information gathering efforts is at precisely the level of merely legal. Companies which are collecting information about the consumer rely on the Web browser to notify the user that an insecure process is taking place. They do not make any separate notice of their own, explaining what information is being gathered or how it is used.

Privacy Protection

The invention does not allow unsolicited interactions with the outside world. In routine operation, the invention has interactions only with the advice servers to the user has subscribed. Assuming that security problems, such as spoofing and moles are not an issue, the risk of compromising privacy is therefore focused on the interaction between consumer and trusted advice provider. As described below, the invention's communications protocol divides the advisory communications process into the following stages:

(ACP-a) Subscription. The consumer anonymously initiates a subscription.

(ACP-b) Gathering. The consumer's advice reader anonymously gathers advice from the site.

(ACP-c) Evaluation. The consumer's advice reader evaluates advice for relevance.

(ACP-d) Explanation. The consumer's advice reader displays a document authored by the advice provider, explaining why a certain advisory is relevant, and proposing a solution/response.

(ACP-e) Solution/Response. The consumer evaluates the document and, potentially, accepts the proposed solution/response, potentially interacting with the world as a result.

The invention, operating with the AEUP communications protocol, makes steps (ACP-a)-(ACP-d) completely private and localizes the information sharing potential to step (ACP-e).

Operationally, a completely ethical advice provider never seeks to violate the privacy protection of steps (ACP-a)-(ACP-d) of the protocol. In particular, a completely ethical provider never seeks to perform covert identification or surveillance of a consumer community using the invention. There are no efforts to infer from server activity the identity or attributes of any user. There are no efforts to develop tools to infer from network activity the attributes of any user. There are no efforts to use the invention as a pure broadcast advertising medium, creating advisories which make unsolicited contact with all or a very large number of consumers. Any efforts to use the invention to gather information from consumers are based on a questionnaire process at solution time (ACP-e) and come with full prior disclosure to the consumer at explanation time (ACP-d), in easily understandable terms, of the types of information being gathered, of the purposes for which they are being gathered. There are no efforts to use information so received in ways unrelated to the disclosed purpose of the information gathering effort. There are no efforts to use information gathered from such a questionnaire to correlate with future server activity.

In one typical implementation, the invention encourages providers to behave in a completely ethical way. The invention may provide mechanisms to encourage consumer knowledge of the standards of completely ethical behavior and knowledge of the standards kept by individual providers. The invention contains mechanisms to defeat and discourage criminal attacks on privacy and to defeat and discourage unethical behavior.

In a typical implementation, the invention has several mechanisms to promote and enforce completely ethical behavior.

First, by encouraging subscription to trusted advice sites, the system encourages users to be aware of the quality of a site. One important component of quality is ethical quality.

Second, the Better Advice Bureau provides a mechanism to issue advisories warning against unethical sites. The Better Advice Bureau maintains an openly accessible list of objective causes for counter advisories. This list makes it clear to consumers and providers the types of behavior which result in counter advisories. In this way, providers receive guidance about what constitutes unethical behavior. Those providers wishing to preserve public trust act ethically.

Third, the invention may frustrate attempts to violate the privacy intent of the protocol. As described below, all legal threats to the protocol have effective responses from the invention, and a provider must engage in criminal activity to violate the communications protocol.

Privacy and AEUP

The invention uses a protocol (AEUP) for information exchange over open public networks which imposes a much higher standard of information ethics than the current industry standard. In addition, the protocol protects against certain outright criminal behavior.

The goal of AEUP is that:

Information on the machine stays on the machine.

That is, information about the consumer's computer or its environment which has been accessed by invention is not distributed to outside parties without explicit consent. In physical terms, AEUP provides a one way membrane between the consumer computer and the outside world. During unattended operation:

Information flows in, but no information flows out of the consumer computer.

This design constraint is expressed in four principles:

(PRIV-a) The act of subscription does not divulge the user's identity or attributes.

(PRIV-b) The act of gathering advice does not divulge the user's identity or attributes.

(PRIV-c) The act of evaluating relevance does not divulge the user's identity or attributes.

(PRIV-d) The act of passively viewing a relevant advisory does not divulge the user's identity or attributes.

When operated under AEUP, all automatic unattended operation preserves the privacy of the user's identity and attributes. The following discussion describes the ways in which AEUP and the overall invention process enable (PRIV-a)-(PRIV-d).

<u>(PRIV-a) Privacy in the act of subscription.</u>

Under AEUP, the information that a certain user is subscribing to a certain advice site is known only to the user and to his advice reader. This requires clarification. In common usage, the word subscription implies a sort of registration process by which a user identifies himself to a provider as a subscriber. Under AEUP, there is no such registration process. There is no need for it. Advice is made freely and anonymously available in the same way that Web sites make Web pages available freely and anonymously. The subscription process is an interaction between the user and the user's own advice reader, not between user and some external advice provider. The advice reader operating on the user's computer obtains from the user the selection of advice sites of interest and stores those on the user's computer only as part of a database maintained locally by the subscription manager component of the advice reader. That database controls the evaluation of advice, causing the advice gatherer to gather advice periodically from some sites and not from others. Subscription is a private matter.

(PRIV-b) Privacy in the act of gathering.

Under AEUP, the act of gathering advice does not reveal information that a certain consumer is interested in certain things, or that he has a certain computer configuration.

It may be objected that an advice site can learn about the identity of a subscriber from the fact that the subscriber's advice reader frequently gathers information from the site. However, in typical implementations, the only thing that can be learned from the act of gathering is that a connection to an advice site has been made from a certain IP address. Under current network protocols most consumers have dynamic IP addresses, and so the correlation between IP address and identity is weak, lasting typically a few minutes. Hence, the information in an IP address is generally of little value.

Moreover, consumers with static IP addresses who do not wish to divulge their true IP address may use a proxy server. Proxy servers are a well known tool by which certain IP client-server transactions are replaced by a three-party client-proxy-server interaction, with the proxy requesting data of the server and routing it anonymously to the client. To the server, it appears that the proxy is the client. To the client, it appears that the proxy is the server. There is never any direct contact between the server and client. The server never obtains the identity of the client, *i.e.* its IP number.

The invention, in one implementation, is configured to offer universal proxy service to all users, and the advice reader offers to the user, as one optional means of connection, the use of such a server. In such an implementation, Better Advice Bureau.org or another central authority offers an anonymous advice gathering server which accepts advice gathering requests from users, strips them of return addresses, routes them to advice sites, and forwards the returned information to the user. This mechanism conceals the IP address of the user.

The act of gathering may be thought to divulge information because the gatherer selects only certain documents from among those available at the advice site. This objection is based on a misunderstanding of AEUP. In a typical implementation, the advice gatherer always accesses all documents available at a certain site, which are not already present on the consumer machine. No selection of any kind is performed at gathering time. Relevance is determined only after all the advice has been gathered and stored on the consumer computer. The only correct inference that can be made from the behavior of the advice gatherer is that the consumer has an ongoing subscription to that site.

This approach is very different from currently popular approaches to obtaining relevant information using Internet. In the currently popular approach, the user fills out a form expressing, for example, preferences, characteristics, and system configurations. This form is sent to the server. The server then responds to the consumer in a focused way, based on the information that was contained in the form. This standard process reveals information about the consumer to the server.

In the invention's approach, the consumer's preferences and configurations are kept confidential on the consumer's machine. All of the advice offered by the site is brought to the consumer machine and is then evaluated for relevance privately.

(PRIV-c) Privacy in the act of evaluating relevance.

The relevance or irrelevance of a given piece of advice can signal a great deal of information about an advice consumer's computer and its environment. A very narrowly focused condition, specifying contents of the user profile, and contents of specific files can, if true, convey a great deal of information about the user.

If the advice reader allows the fact of relevance or irrelevance of an advisory to leak out of the reader to the outside world, it compromises the consumer's privacy. If this happens during unattended operation, the outcome might be very serious because many thousands of advisories are being evaluated for relevance. If there is a mechanism for systematically discovering the relevance of an arbitrary collection of many pieces of advice, a complete profile about the consumer and his environment leaks out.

In a typical implementation, the advice reader's relevance evaluation process has as its only externally observable effect a resulting change in the state of the user interface. The user is notified when a certain piece of advice has become relevant, and that is all. In a typical implementation, the simple fact that something evaluated to relevant causes no activity outside of the user's computer which can be observed by others. There is a possible exception to this when remote inspectors are available. See below.

(PRIV-d)]The act of passively viewing a relevant advisory does not divulge the users identity or attributes.

Reading a text file in the privacy of one's own interaction with one's own computer does not offer any breach of privacy. No one in the outside world need know that one has read the file. However, reading a Web page is a different matter. A hole in the one-way privacy membrane maintained by invention is opened by the careless offering of HTML or other hyperlinked media as a valid type of advisory content in the explanatory component of the advisory. The discussion below describes the hole and its consequences, and describes why the invention, in a typical implementation, does not leave this hole open.

Constraints on Solution Operations

The final step in the advice processing chain is the application of a recommended solution operation. Because this operation can be an essentially arbitrary operation, it is not possible for the invention to control the effects of this operation. In particular, the recommended operation includes electronic correspondence with the advice author, divulging identity and attributes. For this reason, there is a design constraint:

(PRIV-e) In typical implementations, the advice reader does not apply recommended solution operators automatically. They may only be applied after user approval.

Because of the wide-open nature of solution operators, the consumer plays an important role in protecting his own privacy. The act of applying a recommended solution operation may divulge the consumer's identity or attributes, whether the consumer knows this or not. An unethical advice author can create mole solution operators which, while claiming to do one sort of operation, could in fact be conducting electronic correspondence covertly, without informing the consumer. The consumer should only agree to apply solution operations which come from authors he trusts to behave in an ethical fashion.

Remote inspectors: Plugging Leaks

In one implementation, there is a potential violation of the privacy of the relevance evaluation process, based on the assumption that advice reader allows conditional evaluation of and clauses, and the assumption that relevance clauses may refer to conditions which are verified by making queries to other computers and/or other devices remote from the computer on which the advice reader is running. A careless implementation of a remote inspector creates network activity that is observable to the outside world, and from which activity the value of certain relevance clauses is inferred. Inspectors which cause network activity are by no means central to the invention, and this particular privacy threat therefore affects only certain implementations of the invention. (Compare discussion of Covert Channels in Pfleeger, Security in Computing)

Consider an eavesdropper who would like to learn about the value of a relevance clause R when evaluated for relevance on a certain advice consumer's machine. Suppose that the eavesdropper operates an advice site which is trusted by the consumer and subscribed to by the advice reader, so the eavesdropper can introduce advice onto the machine. Suppose that the eavesdropper knows that the advice reader contains an inspector which, when invoked via clause I, generates network activity across a piece of the Internet under control of the eavesdropper. For example, suppose that the eavesdropper has system level access to a node of the Internet in a direct path between the consumer machine and a destination machine that is queried as a result of a certain inspector call. The eavesdropper is then in a position to program the IP transport logic at the node under his control to take note of the existence of IP traffic between the consumer and the destination.

In this hypothetical situation, the eavesdropper is in a position to author advice asserting R and I and to publish the advice at his advice site. After this advice is gathered by the consumer machine, it is evaluated automatically for relevance.

In one implementation of the advice reader, the evaluation of a clause A and B stops immediately as soon as A is determined to be false because it is not necessary to know the value of B to finish the evaluation of the phrase. As soon as A is determined to be false, the phrase A and B is known to have the value False. This scheme is referred to as conditional evaluation. There are implementations of the advice reader that do not perform conditional evaluation. These schemes always evaluate all subexpressions of an expression before inferring the value of the expression. The decision to use conditional evaluation in an implementation is based on performance considerations. Advice readers using conditional evaluation typically run faster.

Assuming that the advice reader implements conditional evaluation as described above, then the network activity prompted by the clause I only occurs if the clause R evaluates to True. The eavesdropper is in a position to observe this network activity, and hence to infer that clause R evaluates to True. Information about the consumer has leaked out of the consumer's computer due to the relevance evaluation.

In discussing this hypothetical situation, it should be noted that eavesdropping activity of the sort described constitutes a form of electronic stalking and may be illegal. Such situation requires either that the trusted advice author be himself an eavesdropper, engaging in conspiracy with the eavesdropper, or does not act to prevent unauthorized advice from being injected in his name, for example by signing his advice. The advice consumer may protect himself from this threat by subscribing to trustworthy sites only, *i.e.* sites meeting the standard of completely ethical behavior.

The advice consumer may also protect himself from this threat by configuring the advice reader to restrict the domain of allowed relevance checking to a domain where he has physical control. In extreme cases, this means limiting relevance to check conditions verifiable only on the machine where the advice reader is running.

There are presently four mechanisms whereby the advice reader can allow network activity and yet protect against this type of eavesdropping.

- Disallow conditional evaluation of clauses. The advice reader is configured to avoid conditional evaluation. In that event, no information about relevance evaluation is revealed by the existence of observable network activity between consumer and destination.

- Randomly reorder subexpressions for conditional evaluation. In evaluation of a clause A and B, the parser randomly reduces the clause to the equivalent of (& A B) with probability 1/2, and to perform (& B A) with probability 1/2. When this is done, the fact that remote network activity occurs in evaluation of the clause R and I implies that either a fair coin was tossed heads or that a clause R was true. This makes it impossible in a particular instance to determine whether R was actually true for the user in question.

- Always force evaluation of subexpressions involving network activity. The advice reader is configured so that each inspector has an attribute Remote-Activity which is set in case the inspector causes activity off the machine running the inspector. The advice reader, in parsing a relevance clause, identifies those subexpressions which have attribute Remote-Activity and forces evaluation of those subexpressions.

- Decouple network activity from relevance evaluation. Inspectors with the attribute Remote-Activity are constrained to work only on cached data, using queued requests, to a prespecified location or collection of locations. This means that an inspector, when receiving a request for an attribute determinable only remotely, can check a local cache. If the answer is found in the cache, it responds with the answer. If the answer is not found in the cache, the request is placed in the queue for future evaluation. Independently, a process runs according to a fixed schedule, *e.g.* once per day, which communicates with a fixed list of remote machines, and which at that time processes all requests that have been cached in the last day. In this way, relevance evaluation *per se* causes no network activity outside of regularly scheduled activity.

An appropriate combination of these mechanisms can safeguard the privacy of relevance evaluation, even in the indicated context of criminal eavesdropping.

HTML: Plugging Leaks

The final appearance of a typical modern HTML document is the product of several files rather than a single one. The HTML document itself gives a kind of logical skeleton of the display, and an inventory of the textual component, and a collection of links to various graphics and multimedia files, which provide the visual components. In traditional Web browsing practice, a Web browser constructs the rendered image in a series of stages. First the HTML file is gathered and the skeleton of the document is rendered. If the HTML document refers to remotely located multimedia files, then the Web browser begins to gather those files;. After the files arrive, they are used to format and render the final display.

Suppose that an advice provider has authored an advisory containing an HTML file making references to files located on the advice providers server in its explanatory component. Suppose also that the advice reader behaves as a traditional Web browser in rendering HTML. At the moment that the consumer reads the advisory, the underlying graphics files is gathered from the advice server. In other words, there is noticeable activity at the advice server caused by the fact of reading an advisory. If the advisory is irrelevant, the HTML is not rendered and, because the unrendered HTML never leads to a gathering of the multimedia file, the server can infer from this activity that an advisory evaluated to relevant. This constitutes a leak of information through the one way membrane, back from consumer to provider.

A completely ethical advice provider must not take any notice of this activity. However, a merely ethical advice provider could, in principle, exploit this fact to learn something about the consumer population. Indeed, such an advice provider can author an advisory referred to a special multimedia file, pointed to only by this advisory. Counting the number of references to the multimedia file, and dividing by the number of gathers of the advisory itself, one can obtain an estimate of the fraction of the consumer population which exhibited a certain combination of circumstances.

However the invention, in a typical implementation, takes steps to frustrate this sort of activity. Inducing leaks of this kind is considered less than completely ethical because, combined with other unethical behavior, it can compromise individual privacy. It is true that such leaks have an innocent and useful application. As long as no correlation is made between the information leaking back and individual identity, one could argue that the leak can be made to serve a constructive purpose of informing the advice provider about the user population in general. However, the existence of such a leak creates a temptation to perform such a correlation, which leads to serious privacy abuses.

There is another mechanism available by which the invention offers similar feedback to advice providers while protecting individual privacy, *i.e.*

randomized response. To discourage attempts to exploit leaks caused by HTML, a typical implementation of invention can employ one or all of three mechanisms:

- HTML-A Proxy server. By working exclusively through a proxy server, the advice reader can destroy all correlation which might otherwise be visible at the advice site between identity of gatherer and fact of gathering. In effect, the advice reader is requesting the multimedia file from the proxy server rather than the original site. In one implementation, the proxy server caches the multimedia file locally and so serves many requests for the multimedia file while only asking for the file once from the advice site. Advice sites may find this arrangement advantageous because it minimizes the load on their own server. In return, they lose the ability to make population attribute prevalence studies, or to make correlation between identity and attributes.

- HTML-B Immediately gather all multimedia. In one implementation of the invention, the gathering process includes the automatic downloading of all multimedia files referred to in the HTML of an advisory. This works as follows: A preliminary parsing of the advisory leads to a listing of all multimedia files referred to in the HTML source of the explanatory component of the advisory. The advice gatherer gathers those files immediately, ensuring that if the advisory ever becomes relevant, the file is available locally. For this implementation of invention, there is no connection between the fact that a file was gathered and the possibility that a certain advisory may be relevant.

Mechanisms (HTML-A) and (HTML-B) may be used simultaneously. That is, a proxy server may gather advice on behalf of a client, and also all multimedia files referred to in any HTML source contained within that advice. The consumer advice reader initially gets only the advisory files, and not all the multimedia files. At the proper time, the multimedia files are gathered from the proxy server. In this way, there is again no connection between the fact that a file was gathered and the possibility that a certain advisory may be relevant.

- HTML-C Download multimedia at random. In one implementation of the invention, the gathering process includes the random downloading of some multimedia files referred to in the HTML of some advisories. This works as follows: A preliminary parsing of the advisory leads to a listing of all multimedia files referred to in the HTML source of the explanatory component of the advisory. The advice gatherer periodically gathers a few randomly selected files from that list. This ensures that, for any advisory that an advice author publishes, a large fraction of the multimedia files are accessed, not for reasons of relevance, but due to outcomes pure chance experiments. Partially, this ensures that among those customers where an advisory becomes relevant, for many of them the file is already available locally. Under this implementation of the invention, there is no logical connection between the fact that a file is gathered and the possibility that a certain advisory is relevant. Whatever connection there may be is probabilistic and could be made rather weak by appropriate choice of the frequency of random downloading.

Support for Privacy Ethics

There are three meta-principles in the invention which help to enforce information ethics.

- Ethical sites. Consumers should only subscribe to advice sites known to behave in an ethical fashion. Many consumers configure their advice reader to subscribe mainly to advice from large concerns which manufacture goods and services of interest to the consumer. For example, a computer manufacturer, a software publisher, or the provider of Internet service. Subscription to substantial organizations of this type is a reasonably secure practice. Such organizations have an interest in providing trustworthy advice so that they maintain rapport with their consumers. Few risks are posed to advice consumers who subscribe to advice authored by such concerns.

- Clear definition of ethics. The Better Advice Bureau is a fundamental tool for encouraging ethical behavior of authors. All users subscribe to this site. This site compiles counter advice, informing users about unethical sites and about unethical advice which has been circulating. Better Advice Bureau defines a solution operator as unethical if it involves divulging information to the author without first informing the user that information is to be divulged or without informing the user accurately about the nature of the information that is to be divulged. If pieces of mole advice are circulating which behave unethically, and they come to the attention of

Better Advice Bureau.org, it may release counter advisories against them. Hence, the Better Advice Bureau functions in some respects as an privacy protection system for the invention, allowing the correction of unethical situations.

- Clear labeling of side effects. To make the definition of ethical behavior clear, and deviation from ethical behavior clear, the Better Advice Bureau describes a set of labels to be attached to advisories, indicating the potential side effects of solution operators. These labels indicate:

  The critical subsystems which may be affected by the advisory's proposed solution.

  Whether information may be revealed by using the advisory's proposed solution.

  What types of information may be so revealed.

  If information may be revealed, whether it may be used for marketing/mailing.

  If information may be revealed, whether it may be shared with other companies.

Completely ethical behavior demands that advice authors label their advice according to its effects on potential consumers. Better Advice Bureau considers it grounds for a counter advisory if an advisory is mislabeled. Persistent, concerted efforts to misinform are considered by Better Advice Bureau grounds for a site counter subscription advisory.

Alternate Client-Server Interactions

A key component of the invention is the synchronization between consumer and provider site images. This happens according to AEUP. However, there are other embodiments of the basic invention in which synchronization is effected by different means. These are described below.

Anonymous Selective Update Protocol

Under this protocol, the act of subscription and the act of synchronization are both anonymous as in the AEUP. However, the update process is selective rather than exhaustive.

ASUP Definition

Under ASUP, each advisory message is abstracted into a short form consisting of at least a message identifier referring to the original advisory, the relevance clause of the original advisory and, potentially, other information, such as a subject line. Under this protocol, the advice server, in addition to directory messages and whole advisory files, also serves to the advice reader the abstracts of one or many advisories.

Under ASUP, the gathering process changes. The advice reader, instead of ensuring that it has the entire body of each advisory of the advice site, ensures that it has at least the abstract for each message. It does this by issuing requests for all the abstracts of all the advisories that are new since the previous synchronization.

Under ASUP, the advice database changes. The database contains two kinds of entries: full advisories, and advisory abstracts.

Under ASUP, the advice reader schedules relevance evaluation for all the relevance clauses it has obtained, both those clauses contained in full advisories and those clauses contained in abstracts.

Under ASUP, a relevant advisory can trigger a new round of contact between advice reader and advice site. Depending on the configuration, the advice reader, either in anticipation of the user wanting the full advisory or after a direct user request, establishes a connection with the advice site, and requests the bodies of certain advisories.

The result of this protocol is that, whereas the consumer's advice reader accesses and evaluates all the published relevance clauses, it does not download all the published advisories.

Analysis of ASUP

This protocol can be advantageous if the published advisories consume considerably more storage than the abstracted advisories. It saves the consumer time in accessing a large body of advisories and saves the provider time in serving requests. A potential drawback of this protocol is the possibility of compromises of consumer privacy. Under the ASUP protocol, it is conceivable that an advice provider attempts to make inferences about the consumer based on observing the advisory files requested and not requested by the advice reader. If the protocol is implemented exactly as described above, the consumer never requests the entire advisory when the clause is not relevant and always request the entire advisory when the clause is relevant. An advice provider whose intent is to learn information about a specific consumer, in principle, correlates server requests for full advisories with IP addresses from which they came, inferring that requests signify the relevance of the corresponding advisory on the corresponding computer. If the IP address is permanently assigned to a certain consumer computer, the provider in principle correlates such requests with consumer identity. In this way, information about the consumer may leak back to the server.

Privacy Protection Under ASUP

- Random gathering. The potential for information leaks is reduced by having the advice reader request full advisory bodies for some advisories whose relevance clauses are not relevant. This is done by a randomization mechanism. Each full advisory body is requested with a probability p, where p is a specified number.

- Proxy server. The potential for information leaks is reduced by having the advice reader request full advisory bodies via a proxy server, which anonymously forwards advisory body requests to the advice site, and thereby masks to the advice site the identity of the requester. A centralized proxy server, for example located at the Better Advice Bureau or at advisories.com is made available for this purpose.

- Proprietary server. The potential for information leaks is reduced by restricting the supply of server software. If the only server software which works with the invention protocol does not to make correlation between consumers and the advisories they request, and also does not log the requests, and if the users of the server software do not attempt to frustrate the intent of the proprietary protocol by eavesdropping on the server-reader transaction, then there is no disclosure of personal information to the server as a result of ASUP.

The supply of server software can be restricted by modifying the reader/server interaction so that a certain security handshake is mandatory. By using digital encryption technology as part of the security handshake and by restricting access to the appropriate security handshake keys, one restricts access to the ability to build server software.

Prohibitions against eavesdropping on client-server interactions can be enforced contractually. Valid server software may be made available only on condition that recipients do not eavesdrop.

Hence there are several avenues to safeguard privacy under ASUP.

NonAnonymous Exhaustive Update Protocol

In certain settings, the concept of anonymous subscription is not workable, for example because advisories are made available only on a for-pay basis, and the reader/server interaction includes a handshake segment in which the reader must qualify himself as a paying customer. A variant on this scenario is in providing advice to members of a club, where members are not in any narrow sense paying for the advice subscription itself, but need to be members to qualify for the advice.

The non-anonymous exhaustive update protocol (NEUP) is applied in a non-anonymous setting where a subscriber exhaustively updates downloading all new advisories at each synchronization. Under NEUP, the consumer's privacy is protected in the following sense: While the fact of the consumer's subscription is known to the provider, the routine act of gathering advice and evaluating relevance does not reveal information about the consumer to the provider.

NonAnonymous Selective Update Protocol

In certain settings, the concept of anonymous subscription is not workable and the use of exhaustive updating is not workable, either because there is a very large body of potentially relevant advisories to consider or each advisory is rather large in size, and very few of the advisories are likely to be relevant, so consumers and providers are not willing to devote extensive resources to exhaustive updating.

The non-anonymous selection update protocol (NSUP) provides this non-anonymous setting where the advice reader selectively updates, obtaining first abstracted advisories, evaluating relevance, and later downloads relevant advisories.

The NSUP by itself gives the consumer no guarantees privacy from the provider. The fact of the consumer's subscription is known to the provider and the routine act of gathering advice and evaluating relevance reveals to the provider which relevance clauses are True. Under NSUP, there are several mechanisms for helping to protect consumer privacy, *e.g.* randomization, proxy server, and proprietary server.

Alternate Advice Distribution

Centralized Advice Server

In one embodiment, a single centralized site stores the advice offered by many different advice providers, with the different advice sites actually serving as different subdirectories of a single file system. All advice readers operating on consumer computers synchronize their site images by contacting this centralized site and requesting resources, such as advisories, from this site. In practice, the single site actually consists of a collection of computers mirroring each other's functions and contents.

This arrangement has an impact in two areas:

- Privacy. This arrangement prevents providers from learning about the identity or about any relevance attributes of any consumers by insulating consumers from providers. In particular, the ASUP protocol is safe in such a setting, provided the central advice site does not log or analyze reader-server transactions.

- Security. This arrangement limits advice sites to those satisfying certain standards imposed by the central server management by restricting the supply of advice sites, and thereby ensures that advice sites are run by typically responsible organizations.

The centralized site allows advice providers to update the contents of their sites on the centralized server by use of standard methods, such as FTP or related file transfer methods.

<u>Centralized Proxy Server</u>

In one embodiment, a single centralized site is available to act as a Proxy server for all advice readers. There is a widely distributed base of advice sites. However, many users do not go to those sites individually. Instead, they configure their advice reader to get all advisories via the centralized proxy server. This is particularly true of users concerned about privacy violations.

The centralized proxy server caches the advice offered by many different advice providers. Advice readers on consumer computers request the proxy server to make available resources, such as advisories, from certain advice sites. If those resources are available on the proxy site, they are served immediately to the user. If they are not available, the original site is queried for the resources, which are both forwarded anonymously to the user, and also placed in the proxy site cache. The advice site includes a method to signal the centralized proxy site when the original site is changed, indicating that it is time to flush the cache (see Hallam-Baker, Phillip M. (1996) Notification for Proxy Caches, World-Wide-Web Consortium Technical Report, http://www.w3.org/TR/WD-proxy).

This arrangement addresses consumer privacy concerns. By insulating consumers from providers, this arrangement prevents providers from learning about the identity or about any relevance attributes of any consumers. In particular, even the ASUP protocol is safe in such a setting, provided the central advice site does not log or analyze reader-server transactions.

Centralized Anonymous Advice Remailer

In one embodiment, advice distribution operates by the use of Internet e-mail transport, routed through a centralized remailer by the use of anonymous mailing lists.

The advice site architecture discussed above is maintained. However, there is a widely distributed base of advice sites. Many readers do not contact those sites directly. Instead, they get advice by anonymous mail. In this implementation, advice sites e-mail their new advisories to the central remailer site, which in turn e-mails them to a mailing list which is kept confidential, consisting of individuals who have contacted the central site and established a subscription relationship. In this implementation, there is a new form of advisory specially designed for retraction. Advice sites handle retraction of advice by e-mailing retraction advisories to the central remailer site, which in turn e-mails them to the mailing list.

Under this arrangement, the advice reader cooperates with the e-mail reader on the consumer computer and with the consumer's e-mail reader configured to filter advice automatically into a mailbox designated for advice reader access. The advice reader performs site synchronization, not by contacting the original advice site, but instead by interpreting the contents of the mailbox that have arrived since the previous synchronization.

This approach is particularly suited for working with POP3 Internet mail servers. This arrangement is essentially an implementation of the AEUP protocol using e-mail. Neither the fact that a certain consumer has a subscription nor the fact of a certain advisory is relevant is generally available to the advice provider.

Under this arrangement, the one way membrane that AEUP provides is made particularly clear to consumers. Consumers understand that the advice site need not know that they subscribe to the site and that there is never direct IP traffic between the consumer machine and the advice site. They can see, by inspecting the plain text of the mail, that advisories are not coming to them directly from the advice site, but instead are transferred anonymously to them from the centralized advice remailer.

A potential weak spot in this arrangement is the existence of a secret mailing list whose secrecy is compromised. To inspire consumer confidence, it is best that the centralized remailer is operated by a trusted consumer minded authority.

By insulating consumers from providers, this arrangement prevents providers from learning about the identity or about any relevance attributes of any consumer who participates in this arrangement and who do not choose to disclose anything to the providers voluntarily.

USENET Advice Diffuser

In one embodiment, advice distribution operates via USENET news transport.

The advice site architecture described above is maintained. There is a widely distributed base of advice sites. However, many readers do not contact those sites directly. Instead, they get advice by USENET. In this implementation, a whole collection of USENET newsgroups is created, *e.g.* one per advice site. The advice site, from time to time, posts new advisories to USENET, which, in turn, cause the new postings to be distributed worldwide to all machines that operate as newsgroup servers.

Under this arrangement, the advice reader then performs site synchronization, not by contacting the original advice site, but instead using USENET protocols to contact a newsgroup server and access new postings in certain newsgroups.

This arrangement is essentially an implementation of the AEUP protocol using USENET. Neither the fact that a certain consumer has a subscription nor the fact of a certain advisory's being relevant is generally available to the advice provider.

Under this arrangement, the one way membrane that AEUP provides is made particularly clear to consumers. Consumers understand that the advice site need not know that they subscribe to the site and that there is never direct IP traffic between the consumer machine and the advice site. In fact, because the act of receiving news via USENET is anonymous, there is not even a mailing list anywhere and so there is no centralized information base linking them to the advice site.

Software Channels

In possible embodiment, advice distribution operates by the use of what are commonly referred to as channels by push providers, such as Backweb, Marimba, and Pointcast (see Ellerman, Castedo (1997) Channel Definition Format, World-Wide-Web Consortium Technical Report, http://www.w3.org/TR/NOTE-CDFsubmit.html). In another embodiment, advice distribution operates by the use of e-mail mailing lists. In either case, the distribution method is referred to as a channel. The logical relationships are the same. Nothing of importance changes below if every occurrence of the word channel is changed to mailing list.

The advice site architecture discussed above is maintained. There is a widely distributed base of advice sites. However, some readers do not contact those sites directly. Instead, they receive advisories through channels. In this implementation, a whole collection of channels is created, perhaps one per advice site. The advice site from time to time pushes new advisories to its channel which, in turn, causes the new offerings to be distributed worldwide to all machines that subscribe to that channel.

Under this arrangement, the advice reader perform site synchronization by listening for incoming data on the channel, and processing the incoming advisories as they arrive.

This arrangement is essentially an implementation of the NEUP protocol. Under some implementations of channels, the fact that a user has a subscription is known to the content provider. Typically, the fact a certain advisory is relevant is generally unavailable to the advice provider.

Under this arrangement, the one way membrane that AEUP provides is made particularly clear to consumers, if channel providers offer truly one-way channels and explain this to consumers. For example, mailing lists are well understood by consumers to offer what is typically a one-way communication. Consumers understand that communication only becomes two-way when the consumer wishes to initiate contacts in the other direction.

Alternate Mechanisms to Promote Consumer Trust

So far it has been assumed that the primary concerns that a consumer might have about privacy must be solved technologically. The viewpoint has been that it is only possible to protect consumer privacy by developing a system which renders it literally impossible for advice providers to make valid inferences about the relevance of certain advisories to specific consumers. It is an important achievement to be able to insulate consumers in this way. However, this insulation comes at the cost of certain constraints. In addition, some consumers may not be able to accept that there exists a purely technological solution to the privacy problem, and those consumers may suspect that any technological solution inevitably has failings, *i.e.* leaks from time to time. Such consumers worry about what happens if a leak occurs, and are not persuaded by technologist's assurances that no leaks can occur. Such consumers might be more reassured by explicit pledges on the part of advice providers that leaks would not be exploited by the providers.

A way to address consumer concerns about advice provider intentions is to restrict the population of advice providers to just those providers who have signed and who are fulfilling a contract to behave in ways which offer consumers guarantees. This has three components:

- Ethical Standards. A fundamental document is made available providing a well known definition of ethical behavior. Certain advice providers have signed this document and deposited it with a central authority, such as Better Advice Bureau, which publishes the identities of signers.

- User Interface. Users are given an option to restrict interactions just to providers who are known to follow the ethical standards.

- Restriction of Server Privileges. The reader/server interaction is protected by a proprietary handshake mechanism, and access to the appropriate reader/server handshaking secret codes is licensed only to those who have signed the agreement on ethics. There are two natural ways this is done:

  By a centralized server strategy, in which advice readers have their functioning restricted by a handshaking mechanism so that they can only interact with a centralized advice server, serving advice only from those sites known to be obligated to follow ethical standards and known to be in compliance.

Following a proprietary server strategy, in which advice readers can only interact with advice servers having the appropriate handshake, and the handshake is known only to servers at ethically bound advice sites.

In summary, there are some providers who have signed an agreement making a contractual guarantee of privacy to customers. There are some consumers who want to deal only with such providers, and there is a technological mechanism to restrict advice reader access to those providers.

Alternate Relevance Evaluation Models

The General Picture: State Comparison

In effect, a relevance clause is an assertion about the state of a computer or of its environment or of the state and environment of computational devices reachable from the computer. The relevance language provides a way for an author to describe components of the state of a computer. However, there are other ways that components of the state could be described.

The advice reader and the associated inspector libraries give a way to compare a description of the state with the actual state. However, there are other ways that components of the state could be compared with a description.

Community of Watchers

An alternate method of state description might rely on a community of watchers, *i.e.* specialized applications, each potentially with its own unique concerns and architecture, which can analyze specific assertions about the computer or its environment. Such an application is referred to as a watcher.

Consider a file watcher application that watches to see if certain files had appropriate attributes. This application maintains a database of assertions. Each entry names a file or directory, a list of the specified attributes of the object, a specified watching frequency, and a pointer to a message and action associated with failure of the assertion. Examples of specifiable attributes include existence, name, version, size, and checksum. The file system watcher, running continually, at scheduled times, or under user control, goes through its database of assertions and checks that each entry has the asserted status, *e.g.* each file has the specified attributes. If it finds an entry that does not have the required status, then it passes information about the failure of the assertion, along with the message and actions associated with the assertion, to a user interface module. The user interface module, a part of the watcher application, and an application used in common across the whole system, presents to the user information about failure of the asserted condition and relays the associated message and recommended response.

A file watcher application also interprets messages making new assertions about the state, or revokes old assertions. The receipt of such a message causes the file watcher to update its database of assertions to include entries making the new assertions or to delete entries making the revoked assertions. The file watcher itself receives these messages from a messaging module, which is part of the watcher application or an application used in common across the whole system.

A remote author who wants to assert conditions about the consumer computer authors messages intended for the file watcher application according to a published file watcher assertion specifier. This is a database entry homologous to the entries in the database kept by the file watcher, or a textual description of an entry, using a keyword language or other humanly interpretable descriptive device. Such a specifier is packaged for transport across networks or by other digital transfer mechanism. Such a package is distributed to consumer machines by any of the methods enumerated so far, *i.e.* AEUP, ASUP, NEUP, NSUP, e-mail, or channels.

Some potential advantages of this approach include:

- Specialization yielding efficiency. A watcher, because it is specialized, is written to optimize the speed at completing a specialized set of tasks. For example, if a file system watcher has to watch several files in the same directory, it is to do so while making only one directory structure access rather than several, thereby saving disk operations. It is possible to avoid certain operations if it is known what the outcome is based on certain earlier operations. If several different assertions must be tested about the same file, it is possible to make a single file access to get the information about all of them simultaneously. In addition, if the watcher accepts instructions in a predefined format that avoids the need for parsing, it can evaluate assertions more quickly.

- Specialization yielding expressiveness. A watcher, because it is specialized, is written to use a very convenient mode of describing a specialized set of tasks. For example, if a file system watcher accepted expressions in a language, that language is designed to incorporate well proven useful idioms from other systems. Thus, in UNIX, wild cards *, [a-z], ? and related constructs are useful in efficiently describing properties of file systems, for example, in referring to a large collection of files with similar but not identical names. A file system watcher makes use of such a specialized idiom without impacting the design of the interfaces of other watchers in the community of watchers.

- Specialized scheduling algorithms. A watcher, because it is specialized, is written to schedule execution of the specialized task set that it addresses appropriately. For example, a file system watcher operating in continuous watch mode follows a specialized scheduling algorithm which is different from the algorithm used for a system settings watcher. In certain operating systems, for example, the file system itself maintains information about whether files or directories changed, which is used to defer evaluation of assertions because it is known that the state of the assertions has not changed since the previous evaluation.

- Specialization yielding security and privacy. A watcher, because it is specialized, is written to block certain dangerous or revealing assertions.

For example, a file system watcher has various user configurable security and privacy settings, enabling the user to control the access to certain files or elements within files.

The collection of watchers is large. In addition to file system watchers and system settings watchers, files such as serial device watchers, printer watchers, and network watchers are provided.

Community of watchers is the same invention

The community of watchers approach is a variation on the invention. There are two ways to understand this point.

- As an implementation layer. Notice that in the invention, the inspector libraries have their actual implementations carried out by variations of such specific watchers. For example, a file system watcher is built to watch various characteristics of various files. This is then exploited by the advice reader, as follows: File related method dispatches in the advice reader are implemented as queries to the file system watcher. The file system watcher answers each query and records the query in its database of assertions. The next time the same dispatch occurs, the file system watcher uses its specialized caching, scheduling, and optimizations to get the answer more cheaply, where feasible. In this way, the community of watchers is an implementation layer for inspectors and the user interface/messaging software of the community of watchers is the advice reader software.

- As a variant implementation. Another way to see that the community of watchers is a related invention is to notice that the features which seem most attractive about the watcher approach, such as enabling specialized idioms for specialized tasks, are provided under both approaches. The UNIX patterning idioms are implemented by creating a named property of World referred to as located files which accepts UNIX-style patterns as the name-specifier string. The fragment:

```
not exists Located files "*.mat" whose(creator of it is creator "MATLAB")
```

which asks for a file in UNIX notation is provided within the invention's language through an inspector for the plural property located files UNIX-pattern.

Forest of Concerns as an Optimization Strategy

The community of watchers approach to state description articulates the concept of forest of concerns. Each interested author formulates a concern about the state of the consumer computer, these concerns are relayed to the computer, and the state of the computer is continually reviewed and compared with those concerns.

From an efficiency and scheduling viewpoint, it is good to organize the process of state description around the concept of a forest of elementary concerns rather than around the concept of relevance clauses. Many pieces of advice may have as subclauses the exact same phrase, and it is inefficient to evaluate those subclauses independently. For example, consider a pool of five pieces of advice with relevance clauses making assertions about the directory Adobe Photoshop. The first is:

```
exists Folder "Brushes and Patterns" of
     Folder containing Application "Adobe Photoshop 2.5"
```

The second is:

```
exists Folder "Calibration" of
     Folder containing Application "Adobe Photoshop 2.5"
```

The third is:

```
exists Folder "Color Palettes" of
    Folder containing Application "Adobe Photoshop 2.5"
```

The fourth is:

```
 exists Folder "Plug-Ins" of
      Folder containing Application "Adobe Photoshop 2.5"
```

The fifth is:

```
 exists Folder "Third-Party Filters" of
      Folder containing Application "Adobe Photoshop 2.5"
```

In each case, evaluation of the relevance clause requires the evaluation of the phrase folder containing Application "Adobe Photoshop 2.5". In short, these five clauses do the same work five times.

It is possible to organize things differently, with the surface expressions being analyzed into a minimal collection of subexpressions. The collection of these subclauses are then watched in nonredundant fashion. More concretely, a pool of relevance clauses scheduled for joint evaluation is parsed into its forest of associated expression trees. This collection of trees is analyzed into its maximal subtrees. Two subtrees are equivalent if they are literally the same, *i.e.* the same method dispatches are applied to the same arguments, or are rearranged under valid applications of commutativity and associativity to be the same. An expression subtree is the child of another subtree if the associated expression occurs as a first level subexpression of the other associated expression.

A subtree is maximal if either:

(a) it has no parents, or

(b) if it has at least two parents and the parents are inequivalent expressions.

The following illustrates the concept with the pool of five relevance clauses illustrated above. The first parses into:

```
(exists (Folder "Brushes and Patterns"
            (Folder-Containing
               (Application "Adobe Photoshop 2.5")
            )
        )
)
```

The second into:

```
(exists (Folder "Calibration"
            (Folder-Containing
               (Application "Adobe Photoshop 2.5")
            )
        )
)
```

The third into:

```
(exists (Folder "Color Palettes"
          (Folder-Containing
            (Application "Adobe Photoshop 2.5")
          )
        )
    )
```

The fourth into:

```
(exists (Folder "Plug-Ins"
          (Folder-Containing
            (Application "Adobe Photoshop 2.5")
          )
        )
    )
```

The fifth into:

```
(exists (Folder "Third-Party Filters"
          (Folder-Containing
            (Application "Adobe Photoshop 2.5")
          )
        )
    )
```

Here, the five different relevance clauses are inequivalent because they name different properties. The collection of maximal expressions consists of these five expressions, plus one proper subexpression:

```
(Folder-Containing
    (Application "Adobe Photoshop 2.5")
 )
```

A watcher organized around the maximal expressions operate in a nonredundant fashion as follows:

- Parse all expressions in a collection of relevance clauses into expression trees.

- Identify with unique labels those maximal subexpressions which have parents.

- Transform each expression tree into a new tree built from references to its labeled maximal subexpressions.

When evaluating relevance, maintain extra storage, referred to as maximal-subexpression value storage, which records the value of maximal subexpressions for later use. When encountering a reference to a labeled maximal subexpression, first check this storage to see if a value is already recorded. If so, use the stored value. If not, evaluate the subexpression, recording the resulting value in the storage.

In more detail, this works as follows: For the pool of five relevance clauses above, the maximal subexpression:

```
(Folder-Containing
     (Application "Adobe Photoshop 2.5")
 )
```

is associated with position one in maximal-subexpression storage. Transform a typical relevance clause by making appropriate references to this storage. In the case of the first of the relevance clauses this works as follows:

```
(exists (Folder "Brushes and Patterns"
            (Maximal-Subexpression 1
              (quote (Folder-Containing
                     (Application "Adobe Photoshop 2.5")
                   )
              )
          )
      )
 )
```

In summary, a wrapper referred to as Maximal-Subexpression is inserted around the identified maximal subexpression. This wrapper method has a first argument which associates the subexpression to storage index one, and a second argument which is a quoted-expression. This quoted expression is not evaluated prior to the invocation of the wrapper method. Instead it is parsed into an appropriate representation as an unevaluated data structure representing an expression for conditional evaluation which is to be passed to the wrapper method as data. The wrapper method looks at location one to see if a value is stored there. If so, the wrapper method returns that value. If not, the wrapper method asks to evaluate the subexpression which it has been passed. Upon completion of the evaluation, it stores the value in location one of the maximal-subexpression storage.

Suppose that this relevance clause is the first evaluated subexpression in a given advice pool, evaluation of which results in evaluation of the subexpression and recording of the value of the subexpression in position one of the maximal-subexpression storage.

Now consider the second item in the pool, in its transformed form:

```
(exists (Folder "Calibration"
                  (Maximal-Subexpression 1
                     (quote (Folder-Containing
                               (Application "Adobe Photoshop 2.5")
                             )
                      )
                 )
           )
     )
```

Suppose this clause is evaluated after the previous clause. There is no evaluation of the maximal subexpression because the wrapper finds that the subexpressions value is already recorded in storage.

It remains to discuss how one can identify maximal subexpressions in a forest of expression trees. This is obtained by a tree/forest pruning algorithm. Define as a terminal form any method invocation which does not depend on any other method evaluations for its value. Formally, it is either a named property of World (Application "Adobe Photoshop 2.5"), an unnamed property of World (System-Folder), or a constant (string "xxxx"), (Integer 1234).

The algorithm begins by scanning a pool of relevance clauses for all unique terminal forms. It associates to each unique terminal form a list of pointers to all locations in the pool where that form occurs.

The algorithm initializes a database of working subexpression forms as the collection of all terminal forms, *i.e.* to begin with, the working subexpression forms are the terminal subexpression forms. These are marked for evaluation at the next stage.

The algorithm proceeds in stages, each stage transforming the working subexpression forms to a collection of parent forms. The algorithm stops when the working database is empty. At a given stage, it iterates through the collection of all working forms. For each form in the working collection marked for study at this stage, it considers the collection of all parent expressions of that expression. This is available because associated with a form is a list of pointers to its occurrences in the pool.

Among those parent method invocations, it identifies the unique forms, *i.e.* the unique combinations of method name and method arguments which have the given subexpression as a first level subexpression. These unique invocation patterns are referred to as parent forms. If there are no parent forms, the subexpression is deleted from the working database. If there is exactly one parent form, the subexpression is replaced in the working database by its parent form, the parent form being marked for processing only at the next stage, and the pointers to the occurrences of the parent form being properly calculated, using the previously available pointers to the children occurrences.

If there is more than one parent form, then a new maximal form is recognized. It is assigned a maximal-form ID number, and a wrapper transformation is made on each expression that references the form. That is, in all those expressions where the form occurs, a wrapper is inserted around the form according to the recipe:

(Maximal-Subexpression $ID# (quote $$))

where ID# is replaced by the ID number of the identified maximal-form, $$ refers to the occurrence of the maximal-form itself, and the (quote) form is the means of preventing immediate evaluation, as described above.

The working forms database is then expanded to include each unique parent form of the recognized maximal-form, with the newly added items marked for evaluation at the following stage, and with a list of pointers to the occurrences of each parent form in the advice pool.

At the conclusion of this algorithm, there is a collection of transformed expressions in which maximal common subexpressions have been identified and where only nonredundant evaluation is performed.

The reader may wish to verify that the algorithm produces exactly the desired result on the pool of five relevance clauses indicated earlier.

Alternates to Binary Relevance Determination

The invention contemplates a situation where messages arrive and computations are performed to evaluate certain assertions with the general goal of notifying the user about certain associated messages, where the timing, format, and other attributes of the notification, including the decision to notify or not, are influenced by the results of the specified computations. The broader notion of influencing relevance appraisal may be implemented by a slight variation on the system described above.

The invention, in one embodiment, obtains appraisals of relevance according to non-binary criteria. A well formed phrase in the relevance language results in numerical values rather than Boolean values. Boolean True is viewed as equivalent to the numerical value1.0, and the Boolean False as equivalent to the numerical value 0.0. Suppose that certain clauses in a body of advice yield Boolean values, but other clauses yield numbers taking values between 0 and 1. A value between 0 and 1 is interpreted as indicating a degree of relevance that lies intermediate between certain relevance and certain irrelevance. In one embodiment, the user interface presents to the user advisories graded according to degree of relevance, with those having value 1.0 at the top of the list and those having value 0.0 at the bottom. This type of variation, extending Boolean to Real, is well known under the name fuzzy logic.

In a different embodiment, the outcome of relevance determination is a categorical label. In this embodiment, True and False are two labels, and the user interface is keyed to display messages labeled True. However, there are labels, such as Attractive Offer or Chronic Household Situation Needing

Eventual Attention. Such labels result from evaluation of relevance clauses and, depending upon the user interface attached to the invention, such labels lead to different methods of notification or different methods of presentation than other kinds of labels. The implementation of a centralized coordination authority such as advisories.com offers a mechanism for publication and coordination of such labels. The implementation of user side filtering allows the user to associate means of notification to various labels, which means include the possibility of no notification.

In one embodiment of the invention, a layer of extra analysis is inserted between relevance appraisal and user interface. Thus, the result of relevance computation may be filtered based on user preferences and on observation of the user. Thus, the relevance computation, rather than determining uniquely the notification status of messages, influences the notification process. For example, a user side filtering method (see above) whereby a user suppresses the display of certain messages which are nominally relevant may be implemented. In one embodiment, such censoring mechanisms are applied automatically. An advice reader or other application contains a module to observe user behavior and make inferences about user preferences which can drive such censoring mechanisms. Similarly, in one embodiment, prioritization mechanisms are applied automatically. An advice reader or other application contains a module to observe user behavior and make inferences about user priorities, so that among relevant messages those which are more likely to be of interest to the user are displayed earlier or more prominently.

Alternate Message Formats

Alternate to MIME Wrappers

The disclosed preferred embodiment uses MIME, a well known Internet standard, as a means of packaging advisories for transport across the Internet and other digital transport media.

Another well known means for packaging textual information for remote interpretation is the XML language. This language also makes possible hierarchical messaging, and is able to accommodate message components of the types enumerated above.

There are many implementations of the basic arrangement disclosed herein. Whether using well known protocols such as MIME and XML or proprietary protocols, they constitute implementations of the invention.

Substitutes for Three-Part Messaging

The invention is discussed in terms of a three-part message, containing humanly interpretable information, a relevance clause, and computer interpretable information. These three logically connected components need not be packaged in the same physical message. There needs to be only an association between these parts. For example, the ASUP protocol sends abstracts containing only message identifiers and the relevance clause separately from the message body, consisting of explanatory content, software, and references. Under ASUP, relevance evaluation drives a second reader-server interaction, where the associated message body is obtained. In other implementations, an even looser association between relevance clause and content is maintained, where a relevant result initiates exploration of a whole sequence of messages.

Substitutes for Relevance Language

The relevance language is a convenient means of describing the state of a consumer computer and its environment. However, other languages can be modified into forms which enable computed-relevance messaging.

JAVA Model

The JAVA programming language is a well known and widely available tool for specifying computations.

In one embodiment of the invention, the role of the relevance language is played using software tools implemented in the JAVA programming language. Owing to the popularity of JAVA this might find wide acceptance among software developers and other computer professionals.

In the currently understood best method of developing this implementation, a special variant of JAVA, RELEVANCE-JAVA is developed, with its own specialized resources and evaluated by a specialized variant of the JAVA machine. The intent of this special version is to provide some of the same privacy and security characteristics as the relevance language described earlier. RELEVANCE-JAVA supplies three specific features which make it very useful:

- Specialized inspector libraries. Special JAVA objects and classes developed to enable the determination of properties of the consumer computer. These inspect file system, system settings, and related properties of the computer and its environment. This is effected by turning on certain features in the JAVA virtual machine which enable access of machine characteristics.

- Privacy Restrictions. While RELEVANCE-JAVA is able to learn a great deal about the user machine, it does not have the ability to transmit any gathered information back to the author. This is effected by limiting the installed objects and classes and turning off certain features in the JAVA virtual machine.

- Security Restrictions. While RELEVANCE-JAVA is able to learn a great deal about the user machine, it does not have the ability to modify the machine, *i.e.* to modify files and to affect the system settings.

The three part messaging model described above is conducted as follows: One part consists of humanly interpretable explanatory content; one part consists of RELEVANCE-JAVA code specifying conditions under which a message becomes relevant on certain consumer machines; and one part of computer interpretable code, perhaps in a different dialect of JAVA, able to cause effects on the consumer machine after consumer approval.

Visual Basic Model

The Visual Basic programming language is a well known and widely available tool for specifying computations.

In one embodiment of the invention, the role of the relevance language is played using software tools implemented in the Visual Basic programming language. Owing to the popularity of Visual Basic this finds wide acceptance among software developers and other computer professionals.

In the currently understood best method of developing this implementation, a special variant of Visual Basic, RELEVANT-BASIC is developed with its own specialized resources and evaluated by a specialized variant of the Basic interpreter. The intent of this special version is to provide some of the same privacy and security characteristics as the relevance language described earlier. RELEVANT-BASIC supplies three specific features which make it very useful:

- Specialized inspector libraries. Special Visual Basic functions and data types are developed to enable the determination of properties of the consumer computer. These have the ability to inspect file system, system settings, and related properties of the computer and its environment.

- Privacy Restrictions. While RELEVANT-BASIC is able to learn a great deal about the user machine, it does not have the ability to transmit any gathered information back to the author. This is effected by limiting the installed objects and classes and turning off certain features in the BASIC interpreter.

- Security Restrictions. While RELEVANT-BASIC is able to learn a great deal about the user machine, it does not have the ability to modify the machine, *i.e.* to modify files and to affect the system settings.

The three part messaging model is conducted as follows: One part consists of humanly interpretable explanatory content; one part consists of RELEVANT-BASIC code specifying conditions under which a message becomes relevant on certain consumer machines; and one part of computer interpretable code, perhaps in a different dialect of Visual Basic, able to cause effects on the consumer machine after consumer approval.

UNIX Model

The UNIX Shell, in its variant implementations, may be viewed as a scripting language, a well known and widely available tool for examining properties of a file system and specifying computations.

In one embodiment of the invention, the role assigned to the relevance language is instead played by software tools implemented in the UNIX shell and associated UNIX Tools. Owing to the popularity of UNIX in its variant forms, this might find wide acceptance among software developers and other computer professionals.

In the currently understood best method of developing this implementation, a special variant of the UNIX Shell, RELEVANT-Shell is developed with its own specialized resources and evaluated by a specialized variant of the Shell interpreter. The intent of this special version is to provide some of the same privacy and security characteristics as the relevance language described earlier. RELEVANT-Shell supplies three specific features which make it useful:

- Specialized inspector Applications. Special applications are developed to enable the determination of properties of the consumer computer. These have the ability to inspect file system, system settings, and related properties of the computer and its environment. These are known to RELEVANT-Shell.

- Privacy Restrictions. While RELEVANT-Shell is able to learn about the user machine, it does not have the ability to transmit any gathered information back to the author. This is effected by disabling access to certain communications and networking features in the shell interpreter.

- Security Restrictions. While the applications reachable through RELEVANT-Shell are able to learn about the user machine, they do not have the ability to modify the machine, *i.e.* to modify files and to affect the system settings, except through standard mechanisms, such as creating temporary files in standard locations such as tmp and subject to resource metering.

The three part messaging model is conducted as follows: One part consists of humanly interpretable explanatory content; one part consists of RELEVANT-Shell code specifying conditions under which a message becomes relevant on certain consumer machines; and one part of computer-interpretable code, perhaps in a different dialect of Shell or other UNIX-interpretable code, able to cause effects on the consumer machine after consumer approval.

Alternate State Description

The possibility of alternate methods of describing the state of the consumer computer is described above. It is possible to describe the state without using an overall relevance language if one has available a community of watchers, each with their own peculiar interfaces. The relevance language is then replaced by whatever means of expression by which the said application modules are invoked and controlled.

Relevance-Mediated Processes

The description of the invention has taken the stance that the purpose of relevance evaluation is to mediate the decision to notify a consumer about the existence of a message. To that end, the advice reader application functions as a messaging center, and advisories play a role analogous to messages in e-mail, USENET news, and other messaging modalities, in that they are read by the user as part of a user defined schedule. In this viewpoint, the user is a manager of his computer, his property, and his affiliations, and he reads advice which helps him with his concerns in that managerial role.

However, there are other non-managerial settings in which relevance can drive the presentation of information to a consumer as an integral part of certain other processes in which the consumer is engaged.

- Guidance. The consumer is the user of a computer applications program, and relevance based messaging provides guidance to the consumer at the moment before performing a certain action or at the moment after performing a certain action.

- Composition. The consumer is reading a document using a display application on the computer, and relevance based content adaptation shapes the document so that the humanly interpretable message targets directly the characteristics of the reader.

In fact, all such applications are embodiments of the invention. Computed relevance messaging is of value much more broadly than in the managerial mode described above.

Relevance-Guided Computer Interaction

The following is an example showing how an advisory is used to guide a user in the operation of a piece of software.

Consider the following problem: A certain dangerous e-mail message has been obtaining wide distribution. When received by a user with the e-mail program Eudora 4.0, the user sees an innocent looking mail message including an attachment with an invitation to the user to open the attachment. The attachment is actually a maliciously prepared document which, if opened, can cause damage to the user's computer.

The discussion below describes one implementation of relevance based messaging which helps to deal effectively with this situation. Under that implementation, an author writes an advisory which is evaluated for relevance before a user of Eudora opens an attachment. The relevance clause inspects various attributes of the contemplated action and precisely targets an attempt to open an attachment with certain attributes. The advisory then returns text to the mail application which the mail application displays to the user.

In one embodiment, the desired effect may be produced using an inter-application communication framework as follows:

- The mail reader application has a special collection of relevance evaluation events, *i.e.* predefined events which are well known to authors of advisories.

- Whenever one of these events occurs, the mail reader notifies the advice reader of the event via a standard event notification protocol.

- The advice reader maintains event pools, *i.e.* advisories intended for evaluation upon receiving notice of certain events.

- The advice reader evaluates the advisories in an event pool upon receiving notice of the corresponding event.

- The advice reader notifies the user of a relevant message by either:

    Notifying the user of the application directly, employing standard user interface devices of the advice reader; or

    Sending the relevant messages to the mail reader. The mail reader then displays those messages for the user, according to the user interface standards of that application.

The choice between these methods of notification is made under the control of user preferences, author preferences, or application defaults.

This event-driven framework is particularly powerful when:

- The application sending an event signal includes descriptive information about the event. In the mail reader context, the event Eudora About to Open Attachment is accompanied by information about the sender of the mail, information about the name of the attachment file, information about the sender of the mail, and information about attributes of the attachment file.

- The advice reader contains an inspector library which refers to properties furnished by the application, *e.g.* mail sender and file name.

In this context, if someone wants to warn every user receiving mail from king@athens.gr with an attachment named trojan.txt that he should not open the attachment, it is possible to author a relevance clause targeting the advisory to those people about to open such an attachment. The routing of advisories to advice event pools is handled through the header line mechanism of MIME and the message line variations discussed above. A simple header line of the form advice-event-pool:, followed by the name of a predefined advice event, indicates the desired routing.

Relevance-Adapted Communication

The following is an example showing how relevance is used to customize the distribution of a body of information (see Fig. 19):

Consider the following problem: A certain publisher wants to create an electronic document whose content is tailored to the reader, for example because it consists of advertising which is more suitable for some readers than others, or because it consists of technical information which is more suitable for some readers than others. However, an ideal customization requires intimate knowledge of the configuration and details of the consumer's preferences, possessions, and affiliations, information which is not likely to be made available by consumers.

The discussion below describes an implementation of a system using the relevance evaluation components of invention. This implementation allows the publisher to create relevance adapted documents, allowing solution of the problem. The publication is distributed as a digital document containing embedded within it references to many possible variations in content. The selection among possible variants is driven by relevance clauses. The components of the document that actually appear on the users display are those which are selected based on intimate knowledge of the characteristics of the user.

The following is one implementation of such a system: A certain base document processing target format is chosen. Suppose for concreteness this is HTML. A special source format is then defined, consisting of documents. In the present context, this is referred to as PRE-HTML. This source format 194 offers the possibility of arranging many hierarchically nested fragments of modified HTML in a linear order. Each component of such an arrangement is protected by one or more relevance clauses. The components of the source format differ from HTML in that they also offer embedded include expressions from the relevance language.

The advisory author writes the document with relevance clauses and inspector clauses 191. To create a custom document for a specific user, the source format document is transported to the user computer 192, and the document in source format is compiled into a custom target format document 195. The target format document is then processed by the intended target document processing system, producing a display of a customized document 193.

The compilation step is the step where the customization occurs and bears closer examination. As the source document is processed, various components are encountered. Those which are protected by relevance clauses which evaluate to False or at any rate not to True are discarded. They do not appear in the final target format file. Those which are protected by relevance clauses which evaluate to True are retained. They do appear in the final target format file. Each retained component is processed before placement in the target document file. If any include expressions are identified in the file, then those expressions are evaluated, and the results are interpolated into the target document file.

This solves the problem of customized document preparation because the relevance language enables the provider to prepare documents which are customized as if the author had access to detailed intimate knowledge of properties of the consumer's computer and environment, but it does so without the need for the consumer to reveal that intimate information to the provider.

This embodiment of the invention posits a provider with information which is presented to various consumers in precisely defined circumstances, and it uses the relevance guarded messaging model described above. Here, the gatherer, the watcher, and the notifier have different structure than they do in the invention as described above, but at an abstract level their functions are similar. For example, the tool which compiles a source format document into a target format document plays the role of both watcher and notifier in the five-part model discussed above, while the target document processing system plays the role of user interface for the notifier. The role of gatherer is played by whatever system or systems bring the source format document into the consumer environment.

There are privacy considerations in this sort of customized documentation. The use of HTML as a target language, for example, means that there is a possibility of leaks.

Other implementations of relevance driven document customization are possible. For example, one could develop a system in which the source document is not compiled once and for all into a target document in a well known format but, rather the source document is structured for interactive interpretation. The following is an example: A source document consists of many pages of PRE-HTML. Embedded in the source document are conditional compilation blocks protected by relevance clauses, and include expression substitutions using the relevance clauses, as described before. As the viewer goes through the document from page to page, each page is compiled from PRE-HTML to HTML and displayed as needed. Under this model, the user's path through the document is determined only at run time. For example, certain links in the document are relevance protected. The relevance expressions refer to attributes of the environment that are changing as the reader progresses through the document, *i.e.* they are changing because the reader is progressing through the document. For example, a reader is prompted for information as part of his reading of the document and, as a result of the prompt, a site profile variable changes, causing pages visited later in the reading to change as a result.

Remote Access to Personal Information

The invention makes it possible for an advisory author to target situations based on an arbitrary combination of computationally verifiable conditions of the consumer computer and its environment. This environment may include data which may be of a personal nature. To the extent that certain kinds of personal data may be widely assumed to exist in a standard format on a substantial population of personal computers this creates the possibility of the invention being used to advise a substantial population of individuals on issues of a personal nature. Natural applications areas include:

- Personal Finance: If information about individual financial assets is assumed to exist on the consumer computer or in its environment in a standard format on a large collection of consumer computers, then advice authors can provide a large body of individuals timely and relevant advice about their bank account management or about their investment portfolio.

- Personal Health Issues: If information about individual medical records is assumed to exist on the consumer computer or in its environment in a standard format on a large collection of consumer computers, then advice authors can provide a large body of individuals timely and relevant advice about drug interactions, or about interactions between genetic or blood type information and drugs.

This creates an unprecedented opportunity, *i.e.* the ability to offer highly targeted advice without compromising individual privacy. Although the advice author is authoring detailed assertions about the finances or health of the consumer, and although it requires intimate knowledge of sensitive personal information to evaluate those assertions, the system itself is not revealing this information back to the author. The consumer may, in some circumstances, choose to reveal such information after reading a relevant advisory.

Such applications are limited by the need for consumers to capture and maintain accurate data in a standard format about items which concern the consumers and which are accessible in a means well known to advice providers. It would be highly desirable to remove the data management and data input burden under this arrangement, so that consumers are not required to become data managers. In particular, it would be highly desirable for the professional organizations responsible for maintaining accurate data about their customers to be the locus of responsibility for data integrity. For example:

- Pharmacies maintain records about their customers.

- Doctors maintain records about their patients.

- Financial institutions maintain records about their clients.

These actors are paid, in part, for keeping accurate and timely records about their patients, customers, or clients.

It would be highly desirable for consumers to have access to some key information that is maintained for them by the professional organizations with which they are affiliated. For example:

- Instead of a consumer entering into his computer data about his drug prescriptions, it would be desirable for the needed data to be obtainable from the pharmacy automatically on demand by the consumer computer.

- Instead of a consumer entering into his computer data about his stock portfolio and manipulating it daily, it would be desirable for any needed data to be obtained from the financial institution automatically on demand by the consumer computer.

- Instead of a consumer entering into his computer data about his health records and manipulating the data as they change, it would be desirable for any needed data to be obtained from the medical institution automatically on demand by the consumer computer.

The following is a solution to this problem using the invention:

- A standard collection of remote medical records inspectors, remote financial records inspectors, and remote drug prescription inspectors is developed, and their syntax and use is published. These inspectors have both server side components and client side components, to be described later.

- Advice authors write advice concerning various issues associated with such personal information.

- Certain doctors, financial institutions, and pharmacies install server side components at computers in their offices. They advertise to the public the availability of remote information access.

- The consumer who is interested in benefiting from advice written using remote information access approaches the financial institution, doctor, or pharmacy and authorizes participation of his own information in the server software.

- The consumer subscribes to certain advice sites whose advice includes advice making use of the remote inspectors. The subscription is initialized appropriately so that the consumer computers advice reader make use of the information.

- Such advice is periodically evaluated according to the advice pool in which the advice is placed. Evaluation causes the consumer computer to establish connections to remote computers to obtain needed information. For example, the remote drug prescription inspector library on the consumer machine establishes a connection with the pharmacy information server and performs certain queries to check if the consumer has certain problematic prescription combinations.

The following is an example of an advisory that is written using this system: Suppose that a certain pharmaceutical manufacturer provides an antidepressant drug to its patients, and that it is discovered that patients who also use a certain anti-inflammatory may experience difficulties. In practice, one prescription might be due to a psychiatrist and the other by an orthopedist who might not be aware of the patient's other medical prescriptions. The manufacturer authors an advisory referring to the dangerous combination as follows:

exists pharmacy prescription "Xanax" and exists pharmacy prescription "Buterin"

The manufacturer includes a description of the potentially dangerous combination for a message body. When the advice reader on the consumer computer encounters this relevance clause, it contacts the pharmacy server with queries for pharmacy prescription Xanax and pharmacy prescription Buterin. It determines the relevance of the advisory based on this. It notifies the consumer of the situation if it turns out to be relevant.

An important issue in determining the consumer acceptance of this system is the ability of the system to protect consumer privacy. To this end, the interaction between client and server is carefully protected:

- The connection between consumer client and pharmacy server is secured by standard cryptographic means (*e.g.* SSL protocol).

- The identity of the client requesting the information is authenticated by the pharmacy server by standard cryptographic means.

By these devices, the pharmacy server avoids revealing information about a person except to the advice reader on that person's computer. The advice reader on that person's computer does not reveal information so received, at least under ordinary operations.

The following is a convenient interaction protocol for such remote inspectors. In this protocol, it is simple to make the client side software. The client transmits, over a secure link, ASCII strings describing the queries exactly as they are described in the surface language. In the above example, the client transmits pharmacy prescription Xanax." The server parses this using a miniature version of the relevance clause parser evaluator. The server knows that this clause refers to the prescription records of Joseph A. Patient because of the initial authentication work and, using standard database inquiry methods, searches the pharmacy database for an entry indicating that Mr. Patient had a pharmacy prescription to Xanax. The server then returns True or False as an ASCII string, and the client parses this string and returns the corresponding Boolean to the advice reader.

Bi-Directional Communications

An intent of the invention is to allow only one way communication, taking information from advice provider to advice consumer, but not allowing information to leak back from consumer to provider. The phrase one way membrane evokes this.

However, there are numerous situations where this model is restrictive. For example, in certain situations consumers are willing to cooperate with providers, particularly when they receive a benefit from cooperating. An example is when consumers want to get technical support to solve a specific problem which existing advisories do not address. For the sake of solving their problem, they are willing to disclose various pieces of information about their configuration to the solution provider. In other situations, advice consumers subscribing to a certain site are actually employees of the organization which operates the advice site, and so they are willing to share information with that particular advice provider.

Open Bi-Directional Communications

The phrase open bidirectional communications refers to a setting where the invention is run and the communications are typically one way, but occasionally there are processes which feed back information to the advice provider, and the process takes place in the clear with the consumer computer identity explicitly available to the provider.

Questionnaires

In one implementation (see Fig. 20), a particular document type is defined, referred to as a questionnaire 200, containing text together with comments, together with distinguished Include-Expressions. Suppose, that Include-Expressions are delimited by double Dollar Signs as in $$. The Include-Expressions are written in the relevance language, and need not evaluate to True or False. For example, they are string- or integer- valued. Suppose also that comments are preceded by %-signs.

An example questionnaire is:

```
% Data needed by ABC Corporation to
% Diagnose the XYZ Problem
Inventory of User Computer Configuration:
Computer Manufacturer: $$ Manufacturer of Computer $$
Model: $$ Model of Computer $$
OSVersion: $$ version of Operating System $$
RAM: $$ System Ram $$
Disk: $$ size of boot volume $$
```

This questionnaire contains text, such as computer manufacturer, as well as Include-Expressions, such as manufacturer of computer. The intent of the questionnaire is that information about the type of computer and about certain features be collected by the advice reader using its rich library of inspectors.

The following is an example showing how questionnaires are used: A questionnaire such as that above is authored by an advice provider 200 and is inserted inside the solution component of an advisory as a MIME component with distinctive content-type 201. The consumer sees a relevant advisory 202, accompanied by humanly interpretable content. The humanly interpretable content says:

You have the XYZ situation. In order to help you,
we at ABC Corp. need some information about this
situation -- information about your system setting. This information can
be automatically
gathered for you if you'll push the button on the left below.
You'll be given a chance to review the information
and then to approve its transmission to ABC Corp.

Below the advisory are two buttons: one saying Gather information and the other saying Review Request. The first button signifies approval to gather the information; the second button signifies a request to view the source file of the questionnaire and thereby learn more about the provider's request to gather data.

If the user approves 203, the relevance clauses in the questionnaire are evaluated 204, for example using various inspectors 205, 206, and the corresponding results are included in the result where the relevance clauses had been. In the case of the previous example, this process produces:

```
% Data needed by ABC Corporation to
% Diagnose the XYZ Problem
Inventory of User Computer Configuration:
Computer Manufacturer: Toshiba
Model: T1200
OSType: Windows 98
```

OSVersion: 1.0

RAM: 64M

Disk: 2G

The user may be shown the results of the include process and given a chance to inspect the results and to relay the results to the advice provider. In one implementation, the results are presented to the user as part of a mailer window, showing the intended recipient of this information 207, and with a button at the bottom marked Send It 208.

By this device, the relevance language simplifies communications between advice provider and advice consumer, allowing inspectors to gather information needed by the advice provider that is difficult for consumers to gather for themselves. The provider is helped because it quickly and accurately obtains information that may be essential in the technical support process, and the customer is helped because the process removes a burden which he would have had of finding the correct data and of reporting it accurately.

For this method to work it must have consumer acceptance. Consumers are sensitive to the possibility of questionnaire spoofing, where a questionnaire purports to gather information of one kind, *e.g.* CPU type, while actually gathering information about another kind, *e.g.* VISA card number or passwords.

One technique to further consumer acceptance is for a privacy ratings service at a central site to certify questionnaires as being in accord with privacy standards when they are appropriate implementations of the randomized response protocol. Under existing Web protocols (see Khare, Rohit (1997) Digital Signature Label Architecture, The World Wide Web Journal, Summer 1997, Vol. 2, Number 3, pp. 49-64, Oreilly, Sebastopol, CA, http://www.w3.org/DSIG) there is a method for the establishment of ratings services which can reliably certify that certain messages have certain properties. The credibility of such assertions, *i.e.* that they are actually made by the service and not by an impostor, is based on deployment of standard authentication and encryption devices. Applying this technology, a privacy ratings service is established at a central site, *e.g.* Better Advice Bureau.org, to certify that certain questionnaires gather information in a fashion generally accepted as appropriate for the advertised task, and the information is used by the solicitor in a manner to protect individual identity. Advice authors seeking certification of the privacy respecting character of their questionnaires submit those messages to the certification authority, which studies the messages and, at its option, agrees to certify some of those messages as privacy respecting. In one embodiment of the invention, the user interface of the advice reader or similar component is configured to permit questionnaires to be displayed to users only when they have been credibly certified by a trusted privacy ratings service.

Mandatory Feedback

In one embodiment of the invention (see Fig. 21), open two-way communication is possible for the purposes of maintaining a relationship with a certain trusted provider.

This assumes a consumer situation different from the usual invention setting. In this variant setting, certain kinds of advice providers enjoy a special status, for example as employers or contractors, which allows them certain coercive privileges not ordinarily enjoyed by advice providers in other settings. These overlord advice sites 210 publish advisories that are gathered by a reader 211, which then performs a relevance evaluation on the advisory 212. Relevant messages are displayed 213 to the user and the user may approve or deny such action 214 as recommended by the advisory. A feedback path 216 enables user actions to be reported 215 to the overlord advice site

In this embodiment, any of the following options may be exercised:

- Certain advice site subscriptions are mandatory;

- Certain advice cannot be deleted by the user, advice by certain providers is not subject to user scheduling, prioritization, or deprecation;

- Certain advice generates automatic feedback from the user to the provider, concerning some or all of:

  (a) The consumer computer's identity;

  (b) The relevance status of a certain advisory on that computer; and

  (c) The fact that a user has/has not taken a certain recommended solution in a certain advisory.

The feedback is transmitted by e-mail or by other convenient electronic means.

In this setting, a manager of many computers can:

(1) write advisories destined to many machines he is managing;

(2) expect that the machines all receive the advisory; and

(3) expect to receive, in return, information about the relevance and/or solution status of the advice on all those machines.

This set of functions may be implemented by modifying the basic advice reader architecture discussed above (see Fig. 22).

- Advice sites 220 may be given a special overlord status (as discussed above in connection with Fig. 21) by configuring the subscription manager of the advice reader to enable such special status.

- A new message line type, Mandated-Action, is instituted and is used by advice sites with overlord status to label a message component with a special keyword phrase as invoking a certain coercive privilege:

  Not user deleteable labels a message as not deletable by the user through the advice reader user interface 221;

  On relevance 222, Evaluate questionnaire 223 and mail back 224 labels a message as requiring immediate notification 225 of the author via a feedback path 226 upon relevance, the notification involving first processing of a questionnaire filling in the various include fields and second transmitting the information to the author;

  Mail back on user acceptance labels a message as requiring immediate notification of the author upon user accepting a proposed action by selecting the action button of an associated advisory;

Mail back on user refusal labels a message as requiring immediate notification of the author upon user accepting a proposed action by selecting the action button of an associated advisory. The advice reader is modified in the appropriate way to carry out the indicated function when a message with overlord status is received and processed.

Masked Bi-Directional Communications

It is possible to enable bidirectional communications while preserving some degree of privacy protection by masking the identity of the respondent.

Masking Via Anonymous Communications and Privacy Ratings

In one implementation (see Fig. 23), an advice provider 231 obtains detailed information from consumer computers while communicating with consumers anonymously, thus enabling consumers to protect their own privacy. This embodiment of the invention limits the scope of communications so that when messages return to the advice provider:

- Message headers contain no information uniquely identifying the respondent;

- Message bodies themselves contain no information uniquely identifying the respondent; and

- The process has these components:

  An advice provider 231 authors a document such as a questionnaire as described above, for gathering information automatically or an HTML form for gathering information by consumer interview. The user's advice reader 232 gathers this information.

Upon determining relevance 233:

If the document is a questionnaire, the advice reader fills in the appropriate include fields.

If the document is an HTML form, the consumer fills in the appropriate survey questions.

The document is e-mailed to the provider via anonymous routing along feedback paths 235, 236 through a certain centralized site, *e.g.* the Better Advice Bureau, advisories.com, or another site 230 offering identity protection via anonymous remailer or functionally equivalent services.

The final stage of this process removes information about the identity of the consumer, by stripping such identity from the message headers. Consumers are expected to have confidence in the fundamental validity of this approach because they understand that the centralized site has an incentive to protect the integrity of the process.

The consumer himself is responsible for ensuring that the message body is free of identifying information. For example, if the consumer responds to an HTML form asking for his name and address, then he is not protecting his own identity. If the consumer forwards a questionnaire containing identifying information, such as IP address, then he is not protecting his own identity.

In one implementation, the consumer protects his privacy with the help of a privacy ratings service at a central site. Under existing internet protocols (see Khare, Rohit, *Digital Signature Label Architecture*, The World Wide Web Journal, Vol. 2, Number 3, pp. 49-64, OReilly (1997) http://www.w3.org/DSIG) there is a method for the establishment of ratings services which reliably certifies that certain messages have certain properties. The credibility of such assertions, *i.e.* that they are actually made by the service and not by an impostor, is based on deployment of standard authentication and encryption devices. Applying this technology, a privacy ratings service is established at a central site, *e.g.* Better Advice Bureau.org, to certify that certain questionnaires do not contain devices soliciting sensitive information. Advice authors seeking certification of the privacy respecting character of their messages submit those messages to the certification authority which studies the messages and, at its option, agrees to certify some of those messages as privacy respecting. In one embodiment of the invention, the user interface of the advice reader or similar component is configured to permit questionnaires and forms to be displayed to users only when they are credibly certified by the privacy ratings service.

Masking Via Randomized Response

In one implementation, an advice provider obtains detailed information from consumer computers while enabling consumers to protect their own privacy. This embodiment of the invention limits the scope of communications so that when messages return to the advice provider:

- Message bodies themselves contain no information which can be reliably inferred to reflect the true state of the consumer's computer or environment.

In certain embodiments, the technique is supplemented by the use of centralized anonymous communications and centralized privacy certifications.

The process has these components:

- An advice provider authors a document similar to a questionnaire as described above, for gathering information automatically, however obeying additional constraints.

- The advice reader fills in the appropriate include fields, randomly changing the answers, and changing the correct answers to incorrect answers, depending on a random mechanism.

- The resulting document is returned to the author.

In one implementation, the process by which the information is returned is made anonymous. The document is addressed to a certain centralized site, *e.g.* the Better Advice Bureau, or advisories.com, or another site offering identity protection via anonymous remailer or functionally equivalent services. This final stage of this process removes information about the identity of the consumer by stripping such identity from the message headers.

The following discussion describes the concept of randomly changing the answers in more detail: Suppose that only questionnaires with Boolean values are allowed, although more general questionnaires are allowed with extra work. The relevance evaluation component of the advice reader evaluates the Boolean expressions indicated in the include fields. However, it does not always insert the result in the outgoing message. Refer to R as the value obtained by relevance evaluation. Instead of always substituting a representation of R in place of the include field, the advice reader conducts a two stage stochastic experiment. At the first stage, it obtains a random Boolean X from a random number generator, the random Boolean being equally likely to be True of False. The value of X is kept private, and drives a decision at the first stage. In this decision, if X is True, the decision is taken to insert a representation of R in the include field. If X is False, the decision is taken to obtain a second Boolean Y, again equiprobable, and to insert a representation of Y in the include field. As a result, in any specific message, it is impossible to say whether the answer obtained at the relevance evaluation stage (R) is True or False on the basis of that message alone because the reported value is equally likely to be R or Y, and the variable X driving the choice between R and Y is not divulged.

This provides a degree of privacy protection for the consumer.

At the same time, this randomized response communications protocol makes it possible for the questionnaire author to obtain information reliably about the population of users while not revealing information about specific users. If π

denotes the fraction of users in the sample with a certain characteristic, and p denotes the fraction of True responses received, then:

$$E(p) = 1/4 + \pi/2$$

where E(·) denotes mathematical expectation.

From p ≈ E(p) (the law of large numbers), π can be estimated by:

$$\hat{\pi} = 2(p - 1/4).$$

For example, if 61% of the responses are True, one estimates that 72 % = 2(61 %-25 %) of the sample has the given characteristic.

There are extensions of the method to non-Boolean variables and to multiple item responses.

For this method to work it must have consumer acceptance. One technique to further consumer acceptance is for a privacy ratings service at a central site to certify messages as being in accord with privacy standards when they are appropriate implementations of the randomized response protocol. Under existing internet protocols (see Khare, Rohit, *Digital Signature Label Architecture*, The World Wide Web Journal, Vol. 2, Number 3, pp. 49-64, Oreilly (1997) http://www.w3.org/DSIG) there is a method for the establishment of ratings services, which reliably certifies that certain messages have certain properties. The credibility of such assertions, *i.e.* that they are actually made by the service and not by an impostor, is based on deployment of standard authentication and encryption devices. Applying this technology, a privacy ratings service is established at a central site, *e.g.* Better Advice Bureau.org, to certify that certain questionnaires use randomized response techniques appropriately and protect individual identity. Advice authors seeking certification of the privacy respecting character of their messages submit those messages to the certification authority which studies the messages and, at its option, agrees to certify some of those messages as privacy respecting. In one embodiment of the invention, the user interface of the advice reader or similar component is configured to permit questionnaires and forms to be displayed to users only when they have been credibly certified by the privacy ratings service.

Network Management

The following discussion describes two important variations of the basic invention which are useful in problems of network management, *i.e.* management of large networks of computational devices.

Mandatory Advice

In the basic description of the invention, it is assumed that advice is offered as a convenience to a human consumer who acts in a managerial role to read and act appropriately at his option (see Fig. 24).

There are settings where the basic communications model described earlier can be usefully modified so that there is no user review of certain advisories. As an example of one such setting, a network administrator 240 supervises a large network of communicating computational devices, each one in a potentially different and dynamically changing state. The network administrator wants certain devices to perform a certain operation, but does not know which devices those are.

In this setting, it is valuable to have an advice reader program 241 which obtains and reviews 242 advisories, but which automatically applies the indicated solution operator 244 when relevance 243 is determined. This enables the network administrator to write a general advisory targeting many machines but not knowing in advance which machines those turn out to be, and obtain the desired functionality on those machines. A solution or communications log 245 may optionally be mailed back to the network administrator via a feedback path 246.

Examples of scenarios where this functionality is useful include:

- Target all machines whose security settings do not match a certain administrator defined standard. Reimpose the required settings on all such machines.

- Target all machines with a copy of a certain file. On such machines, replace the file with an updated version.

- Target all machines which have less than a certain amount of free space on local disk. On such machines, purge the tmp volume.

Other examples can be supplied, including examples outside the technical support application. For example, in a setting where office appliances are computational devices, network management involves tasks concerning the maintenance and monitoring of assets and their use.

In the currently understood best implementation of this variation, there are several changes to the invention:

- The advice reader is implemented as a faceless application with no user interface component.

- The advice reader typically receives advisories by messaging mechanisms alternative to the usual subscription model, for example by e-mail or other diffusion mechanism.

- The message format omits the humanly interpretable content.

- The message format includes a message component containing a software tool, such as a script or executable binary, or a reference to a software tool, such as a URL or a file system pathname, providing functionality to be invoked automatically in case a certain condition becomes relevant.

Certain features may be included in this variant:

- Security Feature. The advice reader includes an authentication feature to verify the identity of the advice site attempting to exert coercive privilege.

- Bi-directional Communication Feature. The advice reader includes the ability to communicate back to the advice Author when the advice Author requires this, as indicated by a Mandated-Action: message line.

Master-Slave Configuration

In the description of the invention, it is assumed that advice is offered as a convenience to a human consumer, who acts in a managerial role to read and act appropriately at his option. In the description, it is assumed implicitly that the consumer is the manager of a personal computer and its environment.

There are settings where the basic communications model described earlier can be usefully modified to reflect the needs of managers of large collections of computational devices. As an example of one such setting (see Fig. 25), a network administrator 250 supervises a large network of communicating computational devices 251-253, each one in a potentially different and dynamically changing state. The network administrator wants to have an advice reader which functions as a master reader 254, in which each entry he sees in the master user interface summarizes the relevance status of advice on many machines 255, 256 simultaneously. This allows the manager to overview 257, 258 and to make decisions about accepting or rejecting advice on many machines at once.

In this setting, the network administrator's workstation is a master machine and the computational devices he manages are slave machines. It is very desirable to have a master advice reader program running on the master machine and which obtains advisories, and which then communicates with the slave machines, each one running a slave relevance evaluator and slave action implementer, and which summarizes the results of the interaction. These slave relevance evaluators accept messages from the master advice reader. The messages consist of wrapper information and individual relevance clauses. The slaves evaluate the relevance clauses in the environment defined by their machines and transmit the resulting values to the master. The master reader then studies the results so obtained and, according to a special master user interface, presents to the network administrator a summary of master relevant messages. A message is deemed master relevant if the associated relevance clause is true on any slave machine. The network administrator studies the master relevant messages and may accept the proposed actions associated with some of them. When he does so, the master reader communicates with the slave action evaluator on slave machines on which a relevant result is obtained, relaying the recommended action part of the advisory, and indicating that the action should be taken. Each slave action evaluator contacted in this way then applies the indicated solution within the environment provided by that machine.

In this setting, a network administrator subscribes to advice and plays the role of managing the advice process in place of all the users of the slave machines. If a piece of advice, when relevant under the ordinary invention, suggests to a user that certain software should be updated on that user's machine, then the same advice is presented to the network administrator instead when some machine on the network should have an update, and it effectively proposes that the corresponding software on every such machine be updated.

In the currently understood best implementation of this variation, there are several changes to the usual invention model.:

- The slave relevance evaluator and slave action implementor are implemented as faceless applications with no user interface component.

- The slave relevance evaluator and slave action implementor typically receive advisories by messaging mechanisms alternative to the usual subscription model, for example by e-mail or other diffusion mechanism.

- The message format for communications between master reader and slave relevance evaluator omit the humanly interpretable content.

- The message format for communications between master reader and slave action implementor include a message component containing a software tool, such as a script or executable binary, or a reference to a software tool, such as a URL or a file system pathname, providing functionality to be invoked automatically.

In addition, certain variations may be exercised as well. The slave advice evaluator and slave action implementor include cryptographic authentication features to verify the identity of the master attempting to exert coercive privilege.

Owing to the difference in outlook that a network administrator has, the Master user interface has features not ordinarily available in the invention. These include:

- Machine List Display. To display a list of all the machines on which a given advisory is relevant. To decorate this list by including other characteristics of the machines.

- Machine List Filtering. To apply selection mechanisms to the list of relevant machines, allowing to apply the recommended action only to a selected subgroup of machines within the relevant group. Particularly useful is the ability to intersect a list of machines with a predefined list, *e.g.* a list of machines in a certain operational division, a list of machines in a certain location, or a list of machines arising as relevant in some other advisory. It is also important to allow the list of machines to be expanded beyond the relevant machines, allowing both editing by hand or concatenation with some other list of machines, for example a predefined list, or a list of machines relevant for some other advisory.

The logical structure described is that of a single body of advisories evaluated for relevance in a collection of different contexts , where the results in all those different contexts are gathered together in one single master user interface. This logical structure makes sense in other settings. For example, in the example of drug interactions discussed above, the pharmacist is an administrator, the body of advisories that he has received from pharmaceutical manufacturers are a body to be applied in many different contexts, and each of his customers database records provide a unique context for interpretation of the advisories. Here, the context is not of individual machines but individual records in a database. The master user interface is the basis for another variation of the invention, *i.e.* operating with a specialized database inspector, the master advice reader obtains a list of all the patients for each advisory for whom a given advisory is relevant. The user interface displays only master-relevant information to the pharmacist, *i.e.* advisories relevant for some patient in the database. The pharmacist then views the relevant advisories and inspects a list of associated patients.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. Accordingly, the invention should only be limited by the Claims included below.